

AD-A043 925

DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2  
MAINTENANCE MANUAL FOR AUDIT. A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)  
AUG 77 R J WYBRANIEC, R REGEN

UNCLASSIFIED

DTNSRDC-77-0075-VOL-4

NL

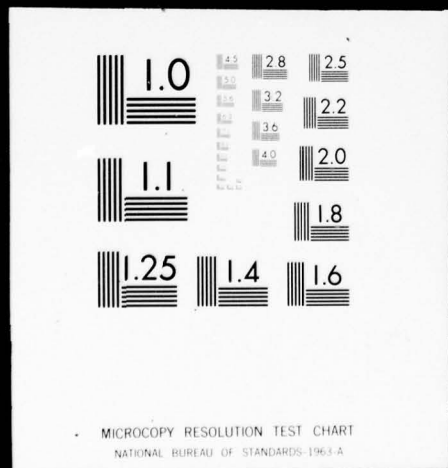
1 of 2  
ADA043925





1 OF 2

ADA043925



ADA043925

NO FILE COPY



DDC FILE COPY

MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE  
VOLUME 4 - APPENDIX D - LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360

AD A 043925

Report 77-0075

# DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084



*12*  
*NW*

## MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE VOLUME 4 APPENDIX D LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360

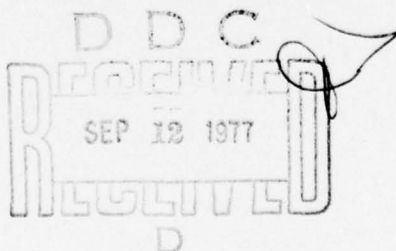
by

Robert J. Wybraniec  
Richard Regen

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

BEST AVAILABLE COPY

COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT  
RESEARCH AND DEVELOPMENT REPORT

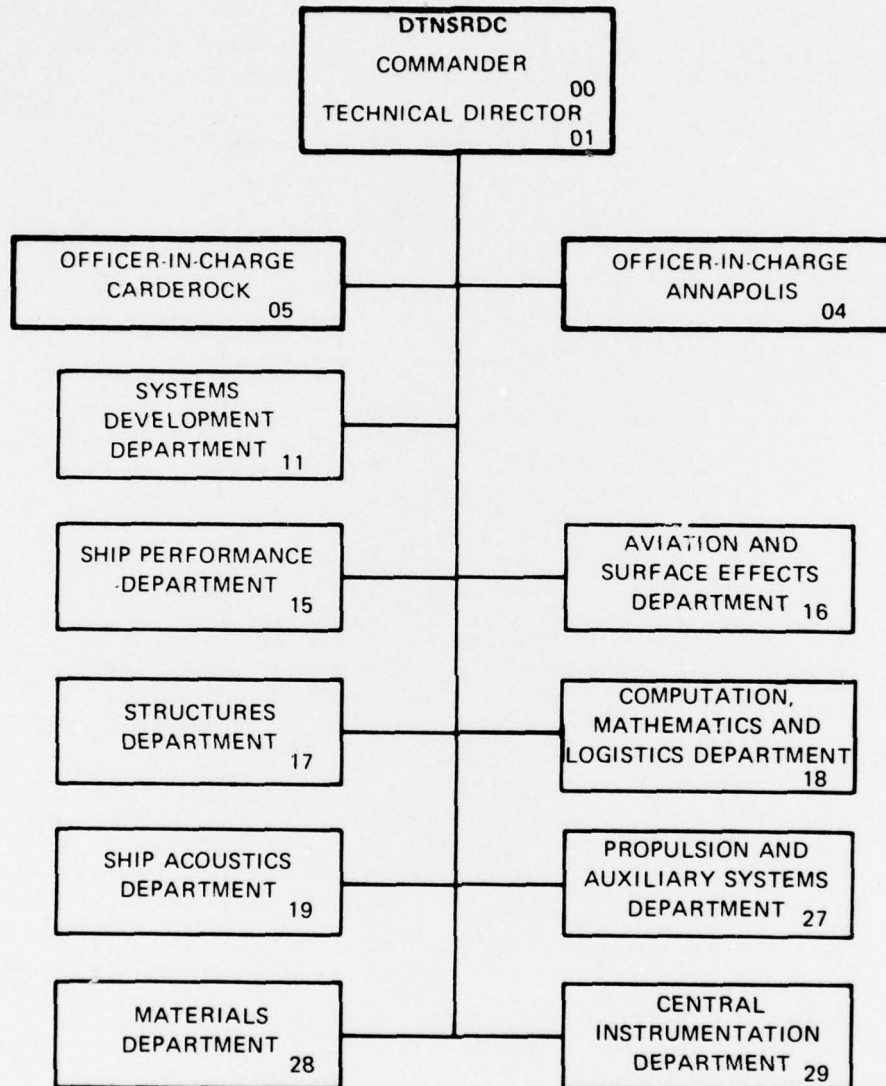


August 1977

Report 77-0075



# MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS





UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) DTNSRDC-77-0075-Vol-4	2. GOVT ACCESSION NO. (9)	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE, VOLUME 4: APPENDIX D • LISTINGS OF THE AUDIT SOFTWARE FOR THE IBM 360		5. TYPE OF REPORT & PERIOD COVERED Final $\neq$ Rept. 1
7. AUTHOR(s) (10) Robert J. Wybraniec Richard Regen		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Navy Surface Effect Ships Project (PMS 304) P.O. Box 34401 - Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (See reverse side)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 176p.		12. REPORT DATE (11) August 1977
		13. NUMBER OF PAGES 177
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) (16) SSH 15, S0308		
18. SUPPLEMENTARY NOTES (17) SSH 15001, S0308001		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) SESCOMP, SESCOMPSPec's, Software Verification, Software Engineering, Reliability, Graph Theory, FORTRAN Software, Modules, Flow Analysis, Variable Precision Execution, Parser, Roll Call, Portability		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The AUDIT documentation provides the maintenance programmer personnel with the information to effectively maintain and use the AUDIT software. The AUDIT software examines FORTRAN computer programs or modules developed under the SESCOMP system for compliance with certain prescribed standards (SESCOMPSPec's) and produces reports detailing the deviations from those standards. The AUDIT software also examines a program unit to detect and (Continued on reverse side)		

DD FORM 1473  
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387682

JP



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 10)

63534N, 19588,  
SSh15001 and S0308001,  
11837001

(Block 20 continued)

report improper use of undefined variables along the program unit's possible paths. In addition, AUDIT has an option which enables the user to test the effect of changes in word length on the output of computer programs.

This report contains the listings of the AUDIT software for the IBM 360.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



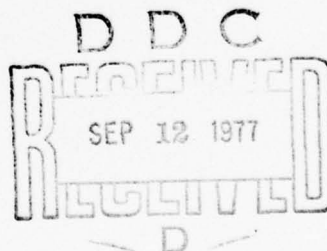
## FOREWORD

The use and maintenance of AUDIT, a software system for analyzing SESCOMP contractor-supplied software, is documented as a set of four separately bound David W. Taylor Naval Ship Research and Development Center volumes sharing the common report number--DTNSRDC 77-0075:

- . Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 1
- . Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 2; Appendix B - Listings of the AUDIT Software for the CDC 6000
- . Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 3; Appendix C - Listings of the AUDIT Software for the UNIVAC 1108
- . Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 4; Appendix D - Listings of the AUDIT Software for the IBM 360

Volume 1 describes AUDIT and the use and maintenance of the AUDIT software. The other three volumes offer software listings for the CDC 6000, UNIVAC 1108, and IBM 360.

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. AND/OR SPECIAL
A	





# INDEX TO PROGRAMS AND SUBPROGRAMS IN APPENDIX D

	<u>Page</u>
AUDIT Main Program	1
MAIN	1
AUDIT Subprograms	5
ARIF	5
ASGOTO	6
ASSIGN	7
AUXIO	8
BITGET	9
BITPUT	9
BLKSTR	10
BUILD	11
CAA	12
CAI	12
CALL	13
CALL2	14
CHKLST	15
CLASS	16
CMPARE	19
CNVRT	20
COM	22
COMCHK	24
COMEXT	27
COMSCH	28
CTGOTO	29
DATA	30
DESCRP	32
DIMEN	33
DO	35
EQUIV	37
ERROR	40
EXPR	46
EXPRCK	48
FLOWCK	49
FNCSTR	52
FORM	54
FORMEL	55
FRMAT	56
GENROL	58
GLOTAB	59
GNLE	60
GOTO	62
GROUP	63
GRT	64
ICOMP	65
IMPTYP	65
INIT	66
INTRIN	68



	<u>Page</u>
IO	69
IOSTR	71
IPREV	71
ITYPE	72
LOGCHK	72
LOGIF	73
LOOPCK	74
LVDLET	75
LVEXIT	78
LVFECH	80
LVFIND	81
LVGRN	83
LVNSRT	84
LVSETP	92
MODID	93
NEXT	94
NXTBLK	94
PARSE	95
PHONEY	100
PRNTS	101
PROG	105
Q1COMP,Q1DPRE,Q1REAL	106
REALCK	110
RECOG	111
RECOV	115
ROLCHK	117
SEARCH	117
SEMANT	118
SEPAR	145
SIMP	146
SLEVEL	147
SQUEEZ	148
SSTOP	149
STATNO	151
STFNC	153
STORE	154
STRCH	154
SJ3	155
SJ3CHK	157
SWITCH	158
SYMTAB	159
TYPE	161
Auxiliary Programs and Associated Data	163
Program GRAPH	163
Syntax Graph	163
PROGRAM SESLIST	169
Basic Interface Definition File	170



# AUDIT Main Program

```

COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000010
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000020
• LOC,ITYP,ITYP,IBLKDT,MODE,IERR,IDES 00000030
DIMENSION IORD(15) 00000040
COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTB(200),EXTTB(100),ISUBS(100) 00000050
COMMON/INPUT/NCALL,IN,IOP 00000060
COMMON/LABELS/STATRA(2,200),NLABEL 00000070
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600) 00000080
COMMON/STFUNC/NSTFNC,ISTFNC(10) 00000090
COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW 00000100
COMMON/HASHLK/IHLOCK(2500),NBLOCK,NB,NBRNCH 00000110
COMMON/NTIMES/NTIMES,IQZ 00000120
COMMON/FLOW/IFL,IRP 00000130
INTEGER A,BLANK,STATRA,BLKTB,EXTTB 00000140
DATA BLANK/1H /,IC/1HC/,IF/1HF/,MAIN/4HMAIN/ 00000150
DATA IORD/29,30,31,32,25,28,19,20,21,22,23,24,26,27,35/ 00000160
IOP=8 00000170
READ(5,12) MODE,IN,IFL,INTR 00000180
12 FORMAT(I1,314) 00000190
NCALL=0 00000200
NTIMES=0 00000210
NREF=0 00000220
NBLK=0 00000230
NSUBS=0 00000240
IRP=IFL-1 00000250
READ(4) NLIST,NINTFC 00000260
READ(4) ((ISUBLT(I,J),I=1,4),J=1,NLIST) 00000270
READ(4) (INTFAC(I),I=1,NINTFC) 00000280
1200 CONTINUE 00000290
4 ISTAT=0 00000300
IOVFLW=0 00000310
MAINPR=0 00000320
NLAREL=0 00000330
NID=0 00000340
JJ=0 00000350
IBLKDT=0 00000360
NSTACK=0 00000370
NHLOCK=0 00000380
ILOOP=0 00000390
NB=0 00000400
IBLKST=0 00000410
NSTFNC=0 00000420
WRITE(6,13) 00000430
13 FORMAT(1H1) 00000440
IFNCNM=0 00000450
DO 2 I=1,11 00000460
DO 2 J=1,500 00000470
2 IDTBL(I,J)=0 00000480
DO 3 I=1,200 00000490
STATRA(1,I)=0 00000500
3 STATRA(2,I)=0 00000510
DO 5 I=1,3 00000520
INITID(I)=0 00000530
5 LASTID(I)=0 00000540
700 CONTINUE 00000550

```



IF(NBLOCK .GT. 2500) GO TO 7000	00000560
CALL BUILD	00000570
WRITE(6,1000) (A(I),I=1,N)	00000580
1000 FORMAT(/6X,100A1,13(/6X,100A1))	00000590
IF(A(1) .EQ. IC) GO TO 700	00000600
IFST=1	00000610
IF(NEXT(IFST) .EQ. BLANK) GO TO 700	00000620
JJ=JJ+1	00000630
CALL CLASS	00000640
IF(ITYP .GT. 18 .AND. ITYP .NE. 28) GO TO 1320	00000650
CALL STATNO	00000660
1320 JPTR=7	00000670
IPREC=ISTAT	00000680
ISTAT=ITYP	00000690
IF(JJ .EQ. 1) GO TO 7	00000700
IF(IBLKDT .EQ. 0) GO TO 6	00000710
IF(ITYP .GE. 18 .AND. ITYP .LE. 27) GO TO 8	00000720
GO TO 220	00000730
6 IF(LTYP .EQ. 9) GO TO 120	00000740
GO TO 8	00000750
7 IPREC=29	00000760
IF(ITYP .GE. 29 .AND. ITYP .LE. 32) GO TO 25	00000770
MAINPR=1	00000780
NXTID(1)=MAIN	00000790
NXTID(2)=BLANK	00000800
25 IF(MAINPR .EQ. 0) GO TO 8	00000810
CALL PROG	00000820
WRITE(IOP,610)	00000830
WRITE(IOP,620)	00000840
8 GO TO (60,70,80,90,100,110,20,140,20,20,40,40,50,50,50,120,130,	00000850
1 200,10,10,10,10,10,170,180,190,200,210,150,38,30,160,220,220,	00000860
2 220,220),ITYP	00000870
10 DO 15 I=1,11	00000880
IF(IPREC .EQ. IORD(I)) GO TO 17	00000890
15 CONTINUE	00000900
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00000910
17 CALL TYPE	00000920
GO TO 500	00000930
20 CALL SIMP	00000940
GO TO 500	00000950
30 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00000960
DO 35 I=1,20	00000970
IF(NEXT(JPTR) .NE. IF) GO TO 35	00000980
JPTR=JPTR-1	00000990
CALL SUB	00010000
GO TO 500	00010010
35 CONTINUE	00010020
38 IF(JJ .NE. 1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00010030
CALL SUB	00010040
GO TO 500	00010050
40 CALL IO	00010060
GO TO 500	00010070
50 CALL AUXIO	00010080
GO TO 500	00010090
60 CALL INIT	00010100



WRITE(6,66) (A(I),I=1,N)	00001110
IF(ITYP.NE.35) GO TO 500	00001120
ISTAT=35	00001130
DO 67 I=1,15	00001140
IF(IPREC.EQ.IORD(I)) GO TO 500	00001150
67 CONTINUE	00001160
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001170
GO TO 500	00001180
70 CALL ASSIGN	00001190
GO TO 500	00001200
80 CALL GOTO	00001210
GO TO 500	00001220
90 CALL ASGOTO	00001230
GO TO 500	00001240
100 CALL CTGOTO	00001250
GO TO 500	00001260
110 CALL ARIF	00001270
GO TO 500	00001280
120 CALL LOGIF	00001290
GO TO 500	00001300
130 CALL DO	00001310
GO TO 500	00001320
140 CALL CALL	00001330
WRITE(6,66) (A(I),I=1,N)	00001340
66 FORMAT(6X,72A1)	00001350
GO TO 500	00001360
150 IF(JJ.NE.1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001370
IBLKOT=1	00001380
CALL SIMP	00001390
GO TO 500	00001400
160 IF(JJ.NE.1) CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001410
CALL PROG	00001420
GO TO 500	00001430
170 DO 175 I=1,12	00001440
IF(IPREC.EQ.IORD(I)) GO TO 177	00001450
175 CONTINUE	00001460
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001470
177 CALL DIMEN	00001480
GO TO 500	00001490
180 DO 185 I=1,5	00001500
IF(IPREC.EQ.IORD(I)) GO TO 187	00001510
185 CONTINUE	00001520
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001530
187 CALL COM	00001540
GO TO 500	00001550
190 DO 195 I=1,13	00001560
IF(IPREC.EQ.IORD(I)) GO TO 197	00001570
195 CONTINUE	00001580
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001590
197 CALL EQUIV	00001600
GO TO 500	00001610
200 DO 205 I=1,14	00001620
IF(IPREC.EQ.IORD(I)) GO TO 207	00001630
205 CONTINUE	00001640
CALL ERROR(2,IDM1,IDM2,IDM3,IDM4)	00001650



207 CALL DATA	00001660
GO TO 500	00001670
210 DO 215 I=1,6	00001680
IF(IPREC.EQ. IORD(I)) GO TO 217	00001690
215 CONTINUE	00001700
CALL ERROR(2, IOM1, IOM2, IOM3, IOM4)	00001710
217 IF(N.GT. 72) GO TO 240	00001720
WRITE(IOP, 218) (A(I), I=1, N)	00001730
218 FORMAT(72A1)	00001740
GO TO 250	00001750
240 WRITE(IOP, 245) (A(I), I=1, N)	00001760
245 FORMAT(72A1/(5X, 1H*, 66A1))	00001770
250 CALL FRMAT	00001780
GO TO 700	00001790
220 CALL ERROR(1, IOM1, IOM2, IOM3, IOM4)	00001800
500 CONTINUE	00001810
IF(N.GT. 72) GO TO 540	00001820
WRITE(IOP, 520) (A(I), I=1, N)	00001830
520 FORMAT(72A1)	00001840
GO TO 600	00001850
540 WRITE(IOP, 545) (A(I), I=1, N)	00001860
545 FORMAT(72A1/(5X, 1H*, 66A1))	00001870
600 IF(MODE.NE. 1) GO TO 700	00001880
IF(ITYP.LT. 30 .OR. ITYP.GT. 32) GO TO 700	00001890
WRITE(IOP, 610)	00001900
610 FORMAT(5X, 15H COMPLEX QICOMP)	00001910
WRITE(IOP, 620)	00001920
620 FORMAT(5X, 24H DOUBLE PRECISION QIDPRE)	00001930
GO TO 700	00001940
2000 WRITE(IOP, 2020)	00001950
2020 FORMAT(6X, 3HEND)	00001960
IF(N.NE. 72) WRITE(6, 2100)	00001970
2100 FORMAT(6X, 22H ILLEGAL END STATEMENT)	00001980
IF(MAINPR.EQ. 1 .OR. IBLKOT.EQ. 1) GO TO 2200	00001990
CALL SUMCHK	00002000
2200 IF(INTR.NE. 1) GO TO 3205	00002010
CALL INTRIN	00002020
3205 CALL SYNTAB	00002030
CALL GWT	00002040
CALL COMCHK	00002050
IF(IOVFLW.EQ. 1) GO TO 3210	00002060
CALL LOOPCK	00002070
IF(IFL.EQ. 0 .OR. IBLKOT.EQ. 1) GO TO 3210	00002080
CALL FLOWCK	00002090
3210 IF(IERR.NE. 2) GO TO 4	00002100
CALL GLDTAB	00002110
IF(MODE.EQ. 1) GO TO 6000	00002120
CALL GENROL	00002130
REWIND 9	00002140
6000 REWIND 8	00002150
STOP	00002160
7000 WRITE(6, 7005)	00002170
7005 FORMAT(/////5X, 54H OVERFLOW OF BASIC BLOCK TABLE - PROCESSING TERM	00002180
*INATED)	00002190
STOP	00002200
END	00002210



# AUDIT Subprograms

```

SUBROUTINE ARIF                                00000010
COMMON A(1326),D(500),IDTRL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID, 00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES 00000040
COMMON/TYP/NQ0,RHSTYP,NQ2,NQ3,LHSTYP 00000050
COMMON/STRING/NTYPE,NSTR,STR(500) 00000060
COMMON/LABELS/STATRA(2,200),NLABEL 00000070
COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH 00000080
INTEGER A,STATRA,STR,COMMA,BLANK,RHSTYP,AY,EF 00000090
INTEGER BITPUT 00000100
DATA LPAR/1H(/,COMMA/1H(/,BLANK/1H /,AY/1H/,EF/1HF/ 00000110
IF(NEXT(JPTR) .NE. AY) GO TO 20 00000120
IF(NEXT(JPTR) .NE. EF) GO TO 20 00000130
IF(NEXT(JPTR) .NE. LPAR) GO TO 20 00000140
JPTR=JPTR-1 00000150
CALL EXPR 00000160
NSTR=NSTR+1 00000170
STR(NSTR)= -5 00000180
NTYPE=1 00000190
CALL PARSE 00000200
CALL FNCSTR 00000210
CALL BLKSTR 00000220
IF(RHSTYP .EQ. 1) CALL ERROR(42) 00000230
NBRNCH=0 00000240
DO 10 I=1,3 00000250
CALL GNLE 00000260
IF(JTYP .NE. 5) GO TO 20 00000270
CALL STSRCH 00000280
STATRA(2,LOC)=BITPUT(STATRA(2,LOC)+1,12) 00000290
IF( NBRNCH .EQ. 0) GO TO 5 00000300
DO 3 J=1,NBRNCH 00000310
IF(LOC .EQ. IBLOCK(NBLOCK-J+1)) GO TO 7 00000320
3 CONTINUE 00000330
5 NBLOCK=NBLOCK+1 00000340
IBLOCK(NBLOCK)=LOC 00000350
NBRNCH=NBRNCH+1 00000360
7 IF(I .EQ. 3) GO TO 10 00000370
IF(NEXT(JPTR) .NE. COMMA) GO TO 20 00000380
10 CONTINUE 00000390
IF(NEXT(JPTR) .NE. BLANK) GO TO 20 00000400
NB=1 00000410
RETURN 00000420
20 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4) 00000430
RETURN 00000440
END 00000450

```



```

SUBROUTINE ASGOTO
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/LABELS/STATRA(2,200),NLABEL
COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
DIMENSION IALPH(4)
INTEGER STATRA,HLANK,COMMA,RPAR,A
INTEGER BITPUT,BITGET
DATA IALPH/1HG,1HO,1HT,1HO/
DATA BLANK/1H /,COMMA/1H /,LPAR/1H(/,RPAR/1H)/
DO 5 I=1,4
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30
5 CONTINUE
CALL GNLE
IF(JTYP .NE. 2) GO TO 30
CALL SEARCH
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(1) .EQ. 1) GO TO 10
IDTYP=1
CALL STORE
LOC=NID
10 CALL IMPTYP
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
$ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
IF(NEXT(JPTR) .NE. COMMA) GO TO 30
IF(NEXT(JPTR) .NE. LPAR) GO TO 30
NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=5000+LOC
NBRNCH=0
20 CALL GNLE
IF(JTYP .NE. 5) GO TO 30
CALL STSRCH
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
IF(NBRNCH .EQ. 0) GO TO 25
DO 22 I=1,NBRNCH
IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 27
22 CONTINUE
25 NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=LOC
NBRNCH=NBRNCH+1
27 IF(NEXT(JPTR) .EQ. COMMA) GO TO 20
IF(A(JPTR-1) .NE. RPAR) GO TO 30
IF(NEXT(JPTR) .NE. BLANK) GO TO 30
NB=1
RETURN
30 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510

```



```

SUBROUTINE ASSIGN
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES
COMMON/LABELS/STATRA(2,200),NLABEL
COMMON/HASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
DIMENSION IALPH(6)
INTEGER BLANK,TEE,OH,STATRA
INTEGER BITPUT,BITGET
DATA BLANK/1H /,TEE/1HT/,OH/1HO/
DATA IALPH/1HA,1HS,1HS,1HI,1HG,1HN/
DO 5 I=1,6
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 20
5 CONTINUE
CALL GNLE
IF(JTYP .NE. 5) GO TO 20
CALL STSRCH
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
IF(NEXT(JPTR) .NE. TEE) GOTO 20
IF(NEXT(JPTR) .NE. OH) GO TO 20
CALL GNLE
IF(JTYP .NE. 2) GO TO 20
CALL SEARCH
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(1) .EQ. 1) GO TO 10
IDTYP=1
CALL STORE
LOC=NID
10 CALL IMPTYP
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
$ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
IF(NEXT(JPTR) .NE. BLANK) GO TO 20
NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=4000+LOC
RETURN
20 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400

```



SUBROUTINE AUXIO	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKOT,MODE,IERR,IDES	00000040
COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000050
DIMENSION IALPH1(6),IALPH2(9),IALPH3(7)	00000060
INTEGER BITGET	00000070
DATA IALPH1/1HR,1HE,1HW,1HI,1HN,1HD/	00000080
DATA IALPH2/1HB,1HA,1HC,1HK,1HS,1HP,1MA,1MC,1HE/	00000090
DATA IALPH3/1HE,1HN,1HD,1HF,1HI,1ML,1HE/	00000100
DATA IBLANK/1H /	00000110
IT=16-ITYP	00000120
IF(IT-2) 25,15,5	00000130
5 DO 10 I=1,6	00000140
IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50	00000150
10 CONTINUE	00000160
GO TO 40	00000170
15 DO 20 I=1,9	00000180
IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50	00000190
20 CONTINUE	00000200
GO TO 40	00000210
25 DO 30 I=1,7	00000220
IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50	00000230
30 CONTINUE	00000240
40 CALL GNLE	00000250
IF(JTYP .NE. 2) GO TO 60	00000260
IF(NEXT(JPTR) .NE. IBLANK) GO TO 50	00000270
CALL SEARCH	00000280
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000290
IF(ISRCH(1) .EQ. 1) GO TO 45	00000300
IDTYP=1	00000310
CALL STORE	00000320
LOC=NID	00000330
45 CALL IMPTYP	00000340
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)	00000350
\$ CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)	00000360
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)	00000370
\$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000380
NBLOCK=NBLOCK+1	00000390
IBLOCK(NBLOCK)=2000+LOC	00000400
RETURN	00000410
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000420
RETURN	00000430
60 CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)	00000440
RETURN	00000450
END	00000460



```

INTEGER FUNCTION BITGET(ILOC,IPOS,IWIDTH)
DIMENSION IMASK(18)
EQUIVALENCE (VAL,IVAL)
DATA IMASK/Z1,Z3,Z7,ZF,Z1F,Z3F,Z7F,ZFF,Z1FF,Z3FF,
* Z7FF,ZFFF,Z1FFF,Z3FFF,Z7FFF,ZFFFF,Z1FFFF,Z3FFFF/
JLOC=ILOC
NSHIFT=32-IPOS
IF(NSHIFT.EQ. 0) GO TO 5
CALL SHIFTR(JLOC,NSHIFT)
5 VAL=AND(JLOC,IMASK(IWIDTH))
BITGET=IVAL
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130

```

```

INTEGER FUNCTION BITPUT(ILOC,IVAL,IPOS)
EQUIVALENCE (RNEW,NEW)
NSHIFT=32-IPOS
JVAL=IVAL
IF(NSHIFT.EQ. 0) GO TO 5
CALL SHIFTL(JVAL,NSHIFT)
5 RNEW=OR(ILOC,JVAL)
BITPUT=NEW
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100

```



```

SUBROUTINE BLKSTR
COMMON A(1326),D(500),IDTHL(11,500),INITID(3),LASTID(3),ISRCH(3),
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
SLOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/FUNC/IFNCRA(5,22),MARG5,IARG5(50),FNCLOC(5),NFUNC
COMMON/LIST/NLIST,NINTFC,ISUHLT(4,200),INTFAC(60)
COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
INTEGER BITPUT,BITGET,FNCLOC
IF(MARG5.EQ. 0) RETURN
DO 100 I=1,MARG5
IOSTAT=2
ICOL=9
LOC=BITGET(IARG5(I),ICOL,9)
NFNC=BITGET(IARG5(I),ICOL+3,3)
IF(NFNC.EQ. 0) GO TO 60
ILOCF=FNCLOC(NFNC)
NARG=BITGET(IARG5(I),ICOL+9,6)
INDEX=BITGET(IDTHL(3,ILOC),32,9)
IF(INDEX.EQ. 0) GO TO 60
IPTR=ISUHLT(4,INDEX)+(NARG-1)/3
JVAR=BITGET(ISUHLT(3,INDEX),14,1)
ICOL=9*MOD(NARG-1,3)+8
IF(JVAR.EQ. 1) ICOL=8
IOSTAT=BITGET(INTFAC(IPTR),ICOL,2)
KPTR=(NARG+5)/3
IEXP=BITGET(IFNCRA(NFNC,KPTR),ICOL+1,1)
IF(IOSTAT.EQ. 2) GO TO 60
IF(IEXP.NE. 0) GO TO 40
IF(IOSTAT.EQ. 1) GO TO 60
GO TO 80
40 IF(BITGET(ISUHLT(3,INDEX),10,4).NE. 0) GO TO 90
INTFAC(IPTR)=BITPUT(INTFAC(IPTR),2,ICOL)
IOSTAT=2
60 NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=2000*LOC
IF(IOSTAT.EQ. 2) GO TO 100
80 NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=1000*LOC
GO TO 100
90 CALL ERROR(55,NARG,IDM2,IDM3,IDM4)
100 CONTINUE
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430

```



```

SUBROUTINE BUILD
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/INOUT/NCALL,IN,IOP
INTEGER A,B,BLANK
COMMON/WASTE/H(72)
DATA BLANK/1H /,ICE/1HC/,IZRO/1H0/
IERR=0
NFIRST=1
NCONTU=0
NCALL=NCALL+1
50 CONTINUE
IF(NFIRST.EQ.1.AND.NCALL.NE.1) GO TO 1
READ(IN,100,END=10) (B(I),I=1,72)
100 FORMAT(72A1)
1 CONTINUE
IF(NFIRST.EQ.1) GO TO 2
IF(B(1).EQ.ICE) GO TO 9
IF(B(6).NE.BLANK.AND.B(6).NE.IZRO) GO TO 6
GO TO 9
2 CONTINUE
DO 3 I=1,72
A(I)=B(I)
3 CONTINUE
NFIRST=0
NCHAR=72
GO TO 50
6 NCONTU=NCONTU+1
IF(NCONTU.LE.19) GO TO 7
IERR=1
CALL ERROR(4,IDM1,IDM2,IDM3,IDM4)
RETURN
7 CONTINUE
DO 8 I=1,66
8 A(NCHAR+I)=B(I+6)
NCHAR=NCHAR+66
GO TO 50
10 IERR=2
9 CONTINUE
N=NCHAR
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430

```



SUBROUTINE CAA(ISTR,MSTR,ID)	00000010
DIMENSION ISTR(6),ID(2),RID(2),JD(2)	00000020
REAL K,IBLANK	00000030
EQUIVALENCE (JD(1),RID(1))	00000040
DATA MASK/ZFF000000/,IBLANK/Z40000000/	00000050
IF(MSTR .GT. 6 .AND. ITYP .NE. 28) CALL ERROR(6,IDM1,IDM2,IDM3,I4)	00000060
JD(1)=0	00000070
JD(2)=0	00000080
J=0	00000090
NCHAR=0	00000100
5 J=J+1	00000110
DO 20 I=1,4	00000120
NCHAR=NCHAR+1	00000130
IF(NCHAR .GT. MSTR) GO TO 10	00000140
K=AND(ISTR(NCHAR),MASK)	00000150
GO TO 15	00000160
10 K=IBLANK	00000170
15 IF(I .EQ. 1) GO TO 20	00000180
NSHIFT=8*(I-1)	00000190
CALL SHIFTR(K,NSHIFT)	00000200
20 RID(J)=OR(RID(J),K)	00000210
IF(J .EQ. 1) GO TO 5	00000220
ID(1)=JD(1)	00000230
ID(2)=JD(2)	00000240
RETURN	00000250
END	00000260

SUBROUTINE CAI(ISTR,MSTR,INTVAL)	00000010
DIMENSION ISTR(10)	00000020
IF(MSTR .GT. 10) GO TO 20	00000030
INTVAL=0	00000040
DO 10 I=1,MSTR	00000050
INT=ISTR(I)	00000060
CALL SHIFTR(INT,24)	00000070
INT=INT-240	00000080
IF(INT .EQ. 0) GO TO 10	00000090
INTVAL=INTVAL+INT*10**(MSTR-I)	00000100
10 CONTINUE	00000110
IF(INTVAL .GT. 2**31-1) GO TO 20	00000120
RETURN	00000130
20 CALL ERROR(3,IDM1,IDM2,IDM3,IDM4)	00000140
RETURN	00000150
END	00000160



```

SUBROUTINE CALL
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/STRING/NTYPE,NSTR,STR(500)
COMMON/LIST/NLIST,NINTFC,ISUHLT(4,200),INTFAC(600)
DIMENSION IALPH(4)
INTEGER HLANK,BITPUT,BITGET
DATA IALPH/1MC,1MA,1ML,1HL/
DATA LPAR/1M(/,HLANK/1M /
DO 5 I=1,4
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 50
5 CONTINUE
IPTR=JPTR
CALL GNLE
IF(JTYP .NE. 2) GO TO 50
CALL SEARCH
IF(ISRCH(1) .EQ. 1) CALL ERROR(24,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(2) .EQ. 1) GO TO 8
IDTYP=2
CALL STORE
LOC=NID
8 CONTINUE
IF(LOC .EQ. IFNCNM) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
ILOC=LOC
NXT=NEXT(JPTR)
IF(NXT .EQ. LPAR) GO TO 17
IF(NXT .NE. HLANK) GO TO 50
IF(BITGET(IDTBL(3,LOC),18,1) .EQ. 1) GO TO 20
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)
DO 10 I=1,NLIST
IF(IDTBL(1,LOC) .NE. ISUHLT(1,I)) GO TO 10
IF(IDTBL(2,LOC) .NE. ISUHLT(2,I)) GO TO 10
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,32)
LISTLC=I
GO TO 22
10 CONTINUE
NLIST=NLIST+1
ISUHLT(1,NLIST)=IDTBL(1,LOC)
ISUHLT(2,NLIST)=IDTBL(2,LOC)
ISUHLT(3,NLIST)=0
ISUHLT(4,NLIST)=0
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,32)
CALL ERROR(52,IDM1,IDM2,IDM3,IDM4)
RETURN
20 LISTLC=BITGET(IDTBL(3,LOC),32,9)
22 CONTINUE
IF(BITGET(ISUHLT(3,LISTLC),6,6) .NE. 0)
$ CALL ERROR(26,IDM1,IDM2,IDM3,IDM4)
RETURN
17 JPTR=IPTR
NTYPE=1
CALL EXPR
CALL PARSE
CALL FNCSTR
CALL BLKSTR
IF(MODE .EQ. 1) GO TO 48
LOC=ILOC
INDEX=BITGET(IDTBL(3,LOC),32,9)
KLAS=BITGET(ISUHLT(3,INDEX),10,4)
IF(KLAS .EQ. 1 .OR. KLAS .EQ. 2) CALL CALL2
RETURN
48 JPTR=IPTR-1
CALL CNVRT
RETURN
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
RETURN
END

```



SUBROUTINE CALL2	00000010
COMMON A(1326),L(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTH,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
INTEGER A,D,COMMA,PPAR	00000050
INTEGER BITPUT,BITGET	00000060
DIMENSION IALPH(13)	00000070
DATA PPAR/1H/,COMMA/1H/,I1/1H1/,IH/1HH/	00000080
DATA IALPH/1HC,1HA,1HL,1HL,1H,1HR,1HO,1HL,1HC,1HH,1HK,1H,1H(/	00000090
DO 15 J=1,13	00000100
K=J+6	00000110
A(K)=IALPH(J)	00000120
15 CONTINUE	00000130
DO 20 I=1,6	00000140
KK=19+4*I	00000150
A(KK-3)=I1	00000160
A(KK-2)=IH	00000170
IWRD=1+(I-1)/4	00000180
IPOS=8*I-32*(IWRD-1)	00000190
IVL=BITGET(IDTBL(IWRD,LOC),IPOS,8)	00000200
A(KK-1)=BITPUT(0,IVL,8)	00000210
IF(I.EQ.6) GO TO 25	00000220
20 A(KK)=COMMA	00000230
25 A(KK)=PPAR	00000240
N=KK	00000250
RETURN	00000260
END	00000270



SUBROUTINE CHKLIST	00000010
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCKM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,INLXDT,MODE,IEER,IDES	00000040
DIMENSION IQUIV(100)	00000050
EQUIVALENCE (IQUIV(1),A(301))	00000060
INTEGER BITGET	00000070
NQUIV=0	00000080
DO 30 I=1,NID	00000090
IDTBL(8,I)=0	00000100
IF (BITGET(IDTBL(3,I),14,1) .EQ. 1) GO TO 20	00000110
IF (BITGET(IDTBL(3,I),16,1) .EQ. 1) GO TO 5	00000120
IF (BITGET(IDTBL(3,I),12,1) .EQ. 1) GO TO 15	00000130
GO TO 10	00000140
5 IDTBL(8,I)=1	00000150
10 IF (BITGET(IDTBL(3,I),17,1) .NE. 1) GO TO 30	00000160
NQUIV=NQUIV+1	00000170
IF (NQUIV .GT. 100) GO TO 60	00000175
IQUIV(NQUIV)=I	00000180
GO TO 30	00000190
15 IF (BITGET(IDTBL(3,I),15,1) .EQ. 0) GO TO 30	00000200
20 IDTBL(8,I)=1	00000210
30 CONTINUE	00000220
IF (NQUIV .EQ. 0) RETURN	00000230
DO 50 J=1,NQUIV	00000240
NXQV=IQUIV(J)	00000250
35 NXQV=IDTBL(10,NXQV)	00000260
IF (NXQV .EQ. IQUIV(J)) GO TO 50	00000270
IF (IDTBL(8,NXQV) .EQ. 0) GO TO 35	00000280
IQV=NXQV	00000290
KTYPE=BITGET(IDTBL(3,IQV),10,3)	00000300
40 NXQV=IDTBL(10,NXQV)	00000310
IF (NXQV .EQ. IQV) GO TO 50	00000320
IF (BITGET(IDTBL(3,NXQV),10,3) .NE. KTYPE) GO TO 40	00000330
IDTBL(8,NXQV)=1	00000340
GO TO 40	00000350
50 CONTINUE	00000360
RETURN	00000370
60 CALL ERROR(94,IDM1,IDM2,IDM3,IDM4)	00000372
RETURN	00000374
END	00000380



```

SUBROUTINE CLASS
COMMON A(1326),D(500),IDTHL(11,500),INITID(3),LASTID(3),ISPC(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IHLKDT,MODE,IEFR,IDES
DIMENSION KALP(48),KSUC(48),KFAL(48),KDEC(10),KF(9)
INTEGER A
DATA KDEC(1),KDEC(2),KDEC(3),KDEC(4),KDEC(5),
1 KDEC(6),KDEC(7),KDEC(8),KDEC(9),KDEC(10)
2 /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
DATA KF(1),KF(2),KF(3),KF(4),KF(5),KF(6),KF(7),KF(8)
1 /1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN/
DATA KBLNK,KLPAR,KRPAR,KEQ/1H,1H(,1H),1H=/
DATA KH,KSLSH,KASTK,KCMA/1HH,1H/,1H0,1H,/
DATA KALP(1),KALP(2),KALP(3),KALP(4) /1H1,1HF,1HN,1HG/
DATA KALP(5),KALP(6),KALP(7),KALP(8) /1H0,1HT,1HO,1H(/
DATA KALP(9),KALP(10),KALP(11),KALP(12) /1HC,1HA,1HO,1HN/
DATA KALP(13),KALP(14),KALP(15),KALP(16) /1HM,1HW,1HP,1HR/
DATA KALP(17),KALP(18),KALP(19),KALP(20) /1HE,1HA,1HO,1HL/
DATA KALP(21),KALP(22),KALP(23),KALP(24) /1HT,1HW,1HF,1HO/
DATA KALP(25),KALP(26),KALP(27),KALP(28) /1HU,1HO,1HI,1HA/
DATA KALP(29),KALP(30),KALP(31),KALP(32) /1H0,1HI,1HW,1HS/
DATA KALP(33),KALP(34),KALP(35),KALP(36) /1HT,1HI,1HE,1HN/
DATA KALP(37),KALP(38),KALP(39),KALP(40) /1HO,1HF,1HX,1HQ/
DATA KALP(41),KALP(42),KALP(43),KALP(44) /1HB,1HA,1HL,1HA/
DATA KALP(45),KALP(46),KALP(47),KALP(48) /1HL,1HP,1HR,1HO/
DATA KSUC(1),KSUC(2),KSUC(3),KSUC(4) / 2, -4, -14, 5/
DATA KSUC(5),KSUC(6),KSUC(7),KSUC(8) / 6, 7, 8, -5/
DATA KSUC(9),KSUC(10),KSUC(11),KSUC(12) / 10, -9, 12, -7/
DATA KSUC(13),KSUC(14),KSUC(15),KSUC(16) / 14, -25, -22, 17/
DATA KSUC(17),KSUC(18),KSUC(19),KSUC(20) / 18, 19, -11, -20/
DATA KSUC(21),KSUC(22),KSUC(23),KSUC(24) / -9, -13, 24, -24/
DATA KSUC(25),KSUC(26),KSUC(27),KSUC(28) / -31, 27, -24, -27/
DATA KSUC(29),KSUC(30),KSUC(31),KSUC(32) / 30, -21, -12, 33/
DATA KSUC(33),KSUC(34),KSUC(35),KSUC(36) / -10, -30, 36, 37/
DATA KSUC(37),KSUC(38),KSUC(39),KSUC(40) / 38, -15, -34, -26/
DATA KSUC(41),KSUC(42),KSUC(43),KSUC(44) / 42, -14, -29, -2/
DATA KSUC(45),KSUC(46),KSUC(47),KSUC(48) / -23, 47, 49, -32/
DATA KFAL(1),KFAL(2),KFAL(3),KFAL(4) / 4, 3, -36, 9/
DATA KFAL(5),KFAL(6),KFAL(7),KFAL(8) / -36, -36, -36, -3/
DATA KFAL(9),KFAL(10),KFAL(11),KFAL(12) / 16, 11, -36, 13/
DATA KFAL(13),KFAL(14),KFAL(15),KFAL(16) / -36, 15, -36, 23/
DATA KFAL(17),KFAL(18),KFAL(19),KFAL(20) / -36, 21, 20, -36/
DATA KFAL(21),KFAL(22),KFAL(23),KFAL(24) / 22, -36, 26, 25/
DATA KFAL(25),KFAL(26),KFAL(27),KFAL(28) / -36, 31, 28, 29/
DATA KFAL(29),KFAL(30),KFAL(31),KFAL(32) / -36, -36, 32, 35/
DATA KFAL(33),KFAL(34),KFAL(35),KFAL(36) / 34, -36, 41, 39/
DATA KFAL(37),KFAL(38),KFAL(39),KFAL(40) / -36, -18, 40, -36/
DATA KFAL(41),KFAL(42),KFAL(43),KFAL(44) / 44, 43, -36, 45/
DATA KFAL(45),KFAL(46),KFAL(47),KFAL(48) / 46, -36, -36, -36/
LTYP=0
IPTH=7
5 CONTINUE
JSAVE=KBLNK
JSW=0
ISW=0

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```



JEQ=0	00000560
JCMA=0	00000570
JHOLL=0	00000580
DO 26 J=IPTR,N	00000590
JCH=A(J)	00000600
IF (JCH .EQ. KRLNK) GO TO 26	00000610
6 IF (JHOLL .LE. 0) GO TO 12	00000620
7 DO 8 L=1,10	00000630
IF (JCH .EQ. KDEC(L)) GO TO 10	00000640
8 CONTINUE	00000650
IF (JHOLL .LE. 1) GO TO 11	00000660
9 IF (JCH .EQ. KH) GO TO 32	00000670
GO TO 11	00000680
10 JHOLL=JHOLL+1	00000690
GO TO 25	00000700
11 JHOLL=0	00000710
12 IF (JCH .EQ. KLPAR) GO TO 20	00000720
13 IF (JCH .EQ. KRPAR) GO TO 18	00000730
14 IF (JCH .EQ. KCMA) GO TO 22	00000740
15 IF (JCH .EQ. KEQ) GO TO 23	00000750
16 IF (JCH .EQ. KSLSM) GO TO 21	00000760
17 IF (JCH .EQ. KASTK) GO TO 21	00000770
GO TO 25	00000780
18 JSW=JSW-1	00000790
IF (JSW .GT. 0) GO TO 25	00000800
19 ISW=1	00000810
GO TO 26	00000820
20 JSW=JSW+1	00000830
21 JHOLL=1	00000840
GO TO 25	00000850
22 IF (JSW) 30,30,21	00000860
23 IF (JSW .GT. 0) GO TO 32	00000870
24 JEQ=1	00000880
25 IF (ISW .GT. 0) GO TO 27	00000890
26 CONTINUE	00000900
GO TO 28	00000910
27 JSAVE=JCH	00000920
JP=J	00000930
28 IF (JEQ .LE. 0) GO TO 32	00000940
29 JTP=1	00000950
GO TO 55	00000960
30 JCMA=1	00000970
IF (JEQ .LE. 0) GO TO 32	00000980
31 JTP=17	00000990
GO TO 55	00010000
32 J=1	00010010
ISW=IPTR	00010020
33 JCH=A(ISW)	00010030
IF (JCH .EQ. KRLNK) GO TO 37	00010040
34 IF (JCH .EQ. KALP(J)) GO TO 36	00010050
35 J=KFAL(J)	00010060
IF (J) 39,39,34	00010070
36 J=KSUC(J)	00010080
IF (J .LE. 0) GO TO 39	00010090
37 ISW=ISW+1	00010100



```

      IF (ISW .LE. N) GO TO 33
38 JCH=KHLNK
      GO TO 35
39 JTP=J
      IF (JTP-3) 55,45,40
40 IF (JTP-6) 55,43,41
41 IF (JTP .LT. 19) GO TO 55
42 IF (JTP - 23) 47,47,55
43 DO 44 L=1,10
      IF (JSAVE .EQ. KDEC(L)) GO TO 55
44 CONTINUE
      LTP=9
      JTP=16
      IPT=JP
      GO TO 5
45 IF (JCMA .LE. 0) GO TO 55
      JTP=4
      GO TO 55
47 L=11
      GO TO 52
48 L=L+1
      IF (L .GT. N) GO TO 55
49 IF (A(L) .EQ. KHLNK) GO TO 48
50 IF (A(L) .EQ. KF(ISW)) GO TO 53
51 IF (ISW .EQ. 1) GO TO 48
52 ISW=1
      GO TO 50
53 ISW=ISW+1
      IF (ISW .LE. M) GO TO 48
54 JTP=31
55 ITYP=JTP
      RETURN
      END

```

```

00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430

```



```

SUBROUTINE CMPARE
DIMENSION IROLCK(2,100),ISUB(2,100)
REWIND 3
NSUB=0
NC=0
DO 5 I=1,100
READ(9,3,END=7) ISUB(1,I),ISUB(2,I)
3 FORMAT(A4,A2)
NSUB=NSUB+1
5 CONTINUE
7 IF(NSUB .EQ. 0) RETURN
DO 40 L=1,12
MODE=L-1
NROLL=0
DO 10 I=1,100
READ(3,3,END=15) IROLCK(1,I),IROLCK(2,I)
NROLL=NROLL+1
10 CONTINUE
15 IF(NROLL .EQ. 0) GO TO 35
DO 30 J=1,NSUB
DO 20 K=1,NROLL
IF(IROLCK(1,K) .EQ. ISUB(1,J) .AND. IROLCK(2,K) .EQ. ISUB(2,J))
* GO TO 30
20 CONTINUE
NC=1
WRITE(6,25) ISUB(1,J),ISUB(2,J),MODE
25 FORMAT(6X,12H SUBROUTINE ,A4,A2,53H WAS NOT CALLED IN THE ROLL CAL
*L MODE FOR MODE INDEX ,I3)
30 CONTINUE
GO TO 40
35 WRITE(6,36) MODE
36 FORMAT(6X,65H NO SUBROUTINES WERE CALLED IN THE ROLL CALL MODE FOR
*L MODE INDEX ,I3)
NC=1
40 CONTINUE
IF(NC .EQ. 0) WRITE(6,50)
50 FORMAT(6X,50H ALL SUBROUTINES WERE CALLED IN THE ROLL CALL MODE)
RETURN
END

```



```

SUBROUTINE CNVRT
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IBLKUT,MODE,IERR,IDES
COMMON/STRING/NTYPE,NSTR,STR(500)
DIMENSION IOP(8),ILOG(2,7),JLOG(3,2),IFUNC1(6),IFUNC2(6),
1 IFUNC3(6)
INTEGER STR,A,D,EQUALS,DECPT
INTEGER BITPUT,BITGET
DATA IOP/1H+,1H-,1H/,1H(,1H),1H.,1H*,1H*/
DATA ILOG/1HL,1HT,1HL,1HE,1HG,1HT,1HG,1HE,1HE,1HQ,1HN,1HE,1HO,1HR/
DATA JLOG/1HA,1HN,1HU,1HN,1HO,1HT/
DATA IFUNC1/1HQ,1H1,1HR,1HE,1HA,1HL/
DATA IFUNC2/1HQ,1H1,1HU,1HP,1HR,1HE/
DATA IFUNC3/1HQ,1H1,1HC,1HO,1HM,1HP/
DATA DECPT/1H./
DO 5 J=1,JPTR
5 D(J)=A(J)
J=JPTR
DO 100 K=1,NSTR
IF(STR(K).GT. 0) GO TO 40
DO 10 I=1,8
IF(STR(K).NE. -I) GO TO 10
D(J+1)=IOP(I)
N3=1
IF(I.NE. 8) GO TO 100
D(J+2)=IOP(I)
N3=2
GO TO 100
10 CONTINUE
DO 15 I=1,7
L=I+8
IF(STR(K).NE. -L) GO TO 15
D(J+1)=DECPT
D(J+2)=ILOG(1,I)
D(J+3)=ILOG(2,I)
D(J+4)=DECPT
N3=4
GO TO 100
15 CONTINUE
DO 20 I=1,2
L=I+15
IF(STR(K).NE. -L) GO TO 20
D(J+1)=DECPT
D(J+2)=JLOG(1,I)
D(J+3)=JLOG(2,I)
D(J+4)=JLOG(3,I)
D(J+5)=DECPT
N3=5
GO TO 100
20 CONTINUE
KL=1
IF(STR(K).EQ. -0) KL=2
IF(STR(K).EQ. -10000) KL=4
IF(STR(K).EQ. -20000) KL=3

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```



GO TO(110,25,30,35),KL	00000560
25 DO 27 I=1,6	00000570
D(J+I)=IFUNC1(I)	00000580
27 CONTINUE	00000590
N3=6	00000600
GO TO 100	00000610
30 DO 32 I=1,6	00000620
D(J+I)=IFUNC2(I)	00000630
32 CONTINUE	00000640
N3=6	00000650
GO TO 100	00000660
35 DO 37 I=1,6	00000670
D(J+I)=IFUNC3(I)	00000680
37 CONTINUE	00000690
N3=6	00000700
GO TO 100	00000710
40 IF(STR(K) .LT. 1000001) GO TO 110	00000720
N3=STR(K)/1.E6	00000730
NLOC=(STR(K)/10000)*10000	00000740
JPTR=STR(K)-NLOC	00000750
KLOC=STR(K)-N3*1.E6	00000760
IF(KLOC .LT. 400000 .OR. KLOC .GT. 500000) GO TO 50	00000770
IHL=(KLOC-400000)/10000	00000772
IF(IHL .EQ. 5) GO TO 60	00000774
DO 45 I=1,N3	00000780
D(J+I)=NEXT(JPTR)	00000790
45 CONTINUE	00000800
GO TO 100	00000810
50 DO 55 I=1,N3	00000820
IWRD=(I-1)/4+1	00000830
IPOS=8*I-32*(IWRD-1)	00000840
ICHAR=BITGET(IDTBL(IWRD,JPTR),IPOS,8)	00000850
55 D(J+I)=BITPUT(0,ICHAR,8)	00000860
GO TO 100	00000861
60 KPTR=JPTR-1	00000862
DO 65 I=1,N3	00000863
65 D(J+I)=A(KPTR+1)	00000864
100 J=J+N3	00000870
N=J	00000880
DO 105 I=1,N	00000890
105 A(I)=D(I)	00000900
RETURN	00000910
110 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)	00000920
RETURN	00000930
END	00000940



```

SUBROUTINE COM
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
DIMENSION IDIM(3),IALPH(6)
INTEGER SLASH,COMMA,BLANK,A,RPAR
INTEGER BITPUT,BITGET
DATA IALPH/1H,1H0,1H1,1H1,1H0,1H1/
DATA SLASH/1H//,COMMA/1H./,BLANK/1H /,RPAR/1H/,LPAR/1H(/
DO 10 I=1,6
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60
10 CONTINUE
IF(NEXT(JPTR) .EQ. SLASH) GO TO 15
JPTR=JPTR-1
12 NXTID(1)=BLANK
NXTID(2)=BLANK
GO TO 20
15 CALL GNLE
IF(A(JPTR-1) .EQ. SLASH) GO TO 12
IF(JTYP .NE. 2) GO TO 60
IF(NEXT(JPTR) .NE. SLASH) GO TO 60
20 CALL COMSCH
IF(ISRCH(3) .EQ. 1) GO TO 25
IDTYP=3
CALL STORE
ICMLOC=NID
GO TO 27
25 ICMLOC=LOC
LSTLOC=IDTBL(9,LOC)
27 CALL GNLE
IF(JTYP .NE. 2) GO TO 60
CALL SEARCH
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(1) .EQ. 1) GO TO 28
IDTYP=1
CALL STORE
LOC=NID
28 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)
$ CALL ERROR(17,NXTID(1),NXTID(2),IDM3,IDM4)
IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1)
$ CALL ERROR(53,NXTID(1),NXTID(2),IDM3,IDM4)
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,16)
ICMSI7=1
IF(NEXT(JPTR) .NE. LPAR) GO TO 40
IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 80
I=0
35 I=I+1
CALL GNLE
IF(JTYP .NE. 5) GO TO 60
IDIM(1)=N2
ICMSI7=ICMSI7*N2
IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
IF(NEXT(JPTR) .EQ. COMMA) GO TO 35
IF(A(JPTR-1) .NE. RPAR) GO TO 60

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```



K=NEXT(JPTR)	00000560
IDTBL(3,LOC)=HITPUT(IDTBL(3,LOC),1,1)	00000570
IF(I.GT. 3) GO TO 60	00000580
IDTBL(3,LOC)=HITPUT(IDTBL(3,LOC),1,7)	00000590
DO 30 J=1,1	00000600
30 IDTBL(4+J,LOC)=IDIM(J)	00000610
40 IF(IDTBL(8,ICMLLOC).EQ. 0) GO TO 45	00000620
IDTBL(8,LSTLOC)=LOC	00000630
GO TO 47	00000640
45 IDTBL(8,ICMLLOC)=LOC	00000650
47 IDTBL(5,ICMLLOC)=IDTBL(5,ICMLLOC)+ICMSIZ	00000660
IDTBL(9,ICMLLOC)=LOC	00000670
IDTBL(9,LOC)=ICMLLOC	00000680
LSTLOC=LOC	00000690
IF(A(JPTR-1).EQ. COMMA) GO TO 27	00000700
IF(A(JPTR-1).NE. SLASH) GO TO 50	00000710
IDTBL(8,LOC)=IDTBL(8,ICMLLOC)	00000720
GO TO 15	00000730
50 IF(NEXT(JPTR).NE. HLANK) GO TO 60	00000740
IDTBL(8,LOC)=IDTBL(8,ICMLLOC)	00000750
RETURN	00000760
60 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000770
RETURN	00000780
80 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000790
RETURN	00000800
END	00000810



```

SUBROUTINE COMCHK
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
$ LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
COMMON/GLOBAL/NBLK,NREF,NSUHS,BLKTB(200),EXTTBL(100),ISUBS(100)
INTEGER BITGET,CMBLK(2,20),TP,SZ,PREVTP,BLKTB
INTEGER SESCOM(2,13),SESERR,SES(2),BLANK
DIMENSION ITPS(6),IORD(6)
DATA IORD/1,2,5,4,3,6/
DATA SESCOM/4HCASE,1H ,3HINA,1H ,3HINB,1H ,3HINC,1H ,3HIOX,1H ,
$ 4HNPAG,1HX,4HLINX,1H ,3HIOY,1H ,4HNPAG,1HY,4HLINY,1H ,3HIOZ,1H ,
$ 4HNPAG,1HZ,4HLINZ,1H /
DATA BLANK/1H /,IBKCOM/2H///
DATA SES/4HSESC,2HOM/
SESERR=0
NSES=0
ICTGW2=0
IBLK=INITID(3)
MODCLS=0
LC2=BITGET(IDTBL(3,1),32,9)
IF(LC2 .EQ. 0 .OR. IBLKDT .EQ. 1) GO TO 1
MODCLS=BITGET(ISUBLT(3,LC2),10,4)
1 IF(IBLK .EQ. 0) GO TO 120
IF(IDTBL(1,IBLK) .EQ. BLANK) GO TO 3
LISTLC=BITGET(IDTBL(3,IBLK),32,9)
KLAS=BITGET(ISUBLT(3,LISTLC),10,4)
ISZ=BITGET(ISUBLT(3,LISTLC),32,15)
IF(IDTBL(5,IBLK) .NE. ISZ) GO TO 70
GO TO 5
3 IF(MODCLS .NE. 1 .AND. MODCLS .NE. 2) GO TO 5
IBNKSZ=BITGET(ISUBLT(3,LC2),32,15)
IF(IDTBL(5,IBLK) .NE. IBNKSZ) CALL ERROR(58,IBKCOM,PLANK,IDM3,IDM4)
5 NBLOC=0
ISUM=0
NTP=0
TP=0
ICOMST=IDTBL(8,IBLK)
LOC=ICOMST
10 PREVTP=TP
IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 15
TP=1
IFST=BITGET(IDTBL(1,LOC),8,8)
IF(IFST .LE. 213 .AND. IFST .GE. 201) TP=4
GO TO 18
15 TP=BITGET(IDTBL(3,LOC),10,3)
18 SZ=1
NDIM=BITGET(IDTBL(3,LOC),7,6)
IF(NDIM .EQ. 0) GO TO 22
DO 20 I=1,NDIM
20 SZ=SZ*IDTBL(4,I,LOC)
22 IF(TP .NE. 2 .AND. TP .NE. 3) GO TO 25
IF(MOD(ISUM,2) .NE. 0) CALL ERROR(64,IDTBL(1,LOC),IDTBL(2,LOC),
$ IDTBL(1,IBLK),IDTBL(2,IBLK))
ISUM=ISUM+SZ

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```



25 ISUM=ISUM+SZ	00000560
IF (IBLKDT .EQ. 1 .AND. IORD(PREVTP+1) .GT. IORD(TP+1))	00000570
\$ CALL ERROR(65, IDTBL(1,IBLK), IDTBL(2,IBLK), IDM3, IDM4)	00000580
IF (KLAS .EQ. 10 .OR. IDTBL(1,IBLK) .EQ. BLANK) GO TO 38	00000590
IF (KLAS .EQ. 9) GO TO 35	00000600
IF (KLAS .EQ. 7) GO TO 40	00000610
ICTGR2=1	00000620
IF (TP .EQ. PREVTP) GO TO 35	00000630
IF (PREVTP .EQ. 0) GO TO 32	00000640
DO 30 I=1, NTP	00000650
IF (TP .EQ. ITPS(I)) GO TO 110	00000660
30 CONTINUE	00000670
32 NTP=NTP+1	00000680
ITPS(NTP)=TP	00000690
35 IF (IBLKDT .EQ. 1) GO TO 38	00000700
IF (HITGET(IDTBL(3,LOC), 21, 1) .EQ. 0)	00000710
\$ CALL ERROR(75, IDTBL(1,LOC), IDTBL(2,LOC), IDM3, IDM4)	00000720
38 LOC=IDTBL(8,LOC)	00000730
IF (LOC .NE. ICOMST) GO TO 10	00000740
GO TO 65	00000750
40 IF (TP .EQ. PREVTP) GO TO 45	00000760
NBLOC=NBLOC+1	00000770
CMBLK(1, NBLOC)=TP	00000780
CMBLK(2, NBLOC)=0	00000790
45 CMBLK(2, NBLOC)=CMBLK(2, NBLOC)+SZ	00000800
IF (IDTBL(1,IBLK) .NE. SES(1)) GO TO 85	00000810
IF (IDTBL(2,IBLK) .NE. SES(2)) GO TO 85	00000820
NSES=NSES+1	00000830
IF (NSES .GT. 13) GO TO 80	00000840
IF (IDTBL(1,LOC) .EQ. SESCOM(1, NSES) .AND. IDTBL(2,LOC) .EQ.	00000850
\$ SESCOM(2, NSES)) GO TO 85	00000860
80 SESEPR=1	00000870
85 LOC=IDTBL(8,LOC)	00000880
IF (LOC .NE. ICOMST) GO TO 10	00000890
IPTR=ISUBLT(4, LISTLC)	00000900
NUPTH=IPTR+NBLOC-1	00000910
KOUNT=0	00000920
NGRP=NBLOC	00000930
DO 50 I=IPTR, NUPTH	00000940
KOUNT=KOUNT+1	00000950
IF (KOUNT .GT. NBLOC) GO TO 65	00000960
SZ=HITGET(INTFAC(1), 17, 17)	00000970
TP=HITGET(INTFAC(1), 20, 3)	00000980
IF (TP .NE. 0) GO TO 48	00000990
CMBLK(2, KOUNT)=CMBLK(2, KOUNT)+SZ	00010000
IF (CMBLK(2, KOUNT) .EQ. 0) GO TO 50	00010010
IF (CMBLK(2, KOUNT) .LT. 0) GO TO 90	00010020
KOUNT=KOUNT-1	00010030
NGRP=NGRP+1	00010040
GO TO 50	00010050
48 IF (CMBLK(1, KOUNT) .NE. TP .OR. CMBLK(2, KOUNT) .NE. SZ) GO TO 90	00010060
50 CONTINUE	00010070
IF (NGRP .NE. HITGET(ISUBLT(3, LISTLC), 6, 6)) GO TO 90	00010080
65 IBLK=IDTBL(4, IBLK)	00010090
GO TO 1	00011000



70 CALL ERROR(58, IDTBL(1, IBLK), IDTBL(2, IBLK), IDM3, IDM4)	00001110
GO TO 65	00001120
90 CALL ERROR(57, IDTBL(1, IBLK), IDTBL(2, IBLK), IDM3, IDM4)	00001130
GO TO 65	00001140
110 CALL ERROR(63, IDTBL(1, IBLK), IDTBL(2, IBLK), IDM3, IDM4)	00001150
GO TO 65	00001160
C**CHECK THAT COMMON BLOCK "SESCOM" IS WELL DEFINED	00001170
120 IF (ICTGR2 .EQ. 0 .AND. (MODCLS .EQ. 1 .OR. MODCLS .EQ. 2))	00001180
\$ CALL ERROR(73, IDM1, IDM2, IDM3, IDM4)	00001190
IF (NSES .EQ. 0) GO TO 130	00001200
IF (SESERR .EQ. 1 .OR. NSES .LT. 13)	00001210
\$ CALL ERROR(48, IDM1, IDM2, IDM3, IDM4)	00001220
RETURN	00001230
130 CALL ERROR(66, IDM1, IDM2, IDM3, IDM4)	00001240
RETURN	00001250
END	00001260



```

SUBROUTINE COMEXT
COMMON A(1326),O(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTH,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
INTEGER BITGET
ICOMLC=0
ICOMNM=IDTBL(9,LOC)
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
INTEGER BITGET
ICOMLC=0
ICOMNM=IDTBL(9,LOC)
ICOMST=IDTBL(8,ICOMNM)
ICOMND=IDTBL(9,ICOMNM)
ICOMSZ=IDTBL(5,ICOMNM)
NXTLOC=ICOMND
DO 20 I=1,ICOMSZ
NXTLOC=IDTBL(8,NXTLOC)
MUL=1
ISZ=1
ITP=BITGET(IDTBL(3,NXTLOC),10,3)
IF(ITP.EQ.2.OR.ITP.EQ.3)MUL=2
IF(LOC.EQ.NXTLOC)GO TO 25
IF(BITGET(IDTBL(3,NXTLOC),1,1).EQ.1)GO TO 10
GO TO 20
10 NDIM=BITGET(IDTBL(3,NXTLOC),7,6)
DO 15 J=1,NDIM
15 ISZ=ISZ+IDTBL(J+4,NXTLOC)
20 ICOMLC=ICOMLC+ISZ*MUL
25 ILFT=ICOMLC
IRHT=ICOMSZ-ICOMLC
NXTLOC=LOC
IOFFST=IDTBL(11,NXTLOC)
30 NXTLOC=IDTBL(10,NXTLOC)
IF(NXTLOC.EQ.LOC)RETURN
IOFF2=IDTBL(11,NXTLOC)
IF((IOFFST-IOFF2).GT.ILFT)GO TO 50
ITP=BITGET(IDTBL(3,NXTLOC),10,3)
ISZ=1
IF(ITP.NE.2.AND.ITP.NE.3)GO TO 32
IF(MOD((ILFT-IOFFST+IOFF2),2).NE.0)CALL ERROR(64,IDTBL(1,NXTLOC),
5),IDTBL(2,NXTLOC),IDTBL(1,ICOMNM),IDTBL(2,ICOMNM))
ISZ=2
32 IF(BITGET(IDTBL(3,NXTLOC),1,1).NE.1)GO TO 40
NDIM=BITGET(IDTBL(3,NXTLOC),7,6)
DO 35 I=1,NDIM
35 ISZ=ISZ+IDTBL(4+I,NXTLOC)
40 IF((ISZ-(IOFFST-IOFF2)).GT.IRHT)GO TO 50
GO TO 30
50 CALL ERROR(47,IDM1,IDM2,IDM3,IDM4)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470

```



SUBROUTINE COMSCH	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
J=INITID(3)	00000050
IF(J.EQ. 0) GO TO 15	00000060
DO 10 I=1,NID	00000070
IF(ICOMP(NXTID,IDTBL,J,11).EQ. 0) GO TO 5	00000080
ISRCH(3)=1	00000090
LOC=J	00000100
RETURN	00000110
5 J=IDTBL(4,J)	00000120
IF(J.EQ. 0) GO TO 15	00000130
10 CONTINUE	00000140
15 ISRCH(3) =0	00000150
RETURN	00000160
END	00000170



SUBROUTINE CTGOTO	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IEHR,IDES	00000040
COMMON/LABELS/STATRA(2,200),NLABEL	00000050
COMMON/HASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000060
DIMENSION IALPH(4)	00000070
INTEGER STATRA,A,BLANK,RPAR,COMMA	00000080
INTEGER BITPUT,BITGET	00000090
DATA BLANK/1H /,COMMA/1H /,LPAR/1H (/ ,RPAR/1H )/	00000100
DATA IALPH/1MG,1MO,1MT,1HO/	00000110
DO 5 I=1,4	00000120
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30	00000130
5 CONTINUE	00000140
IF(NEXT(JPTR) .NE. LPAR) GO TO 30	00000150
NBLOCK=NBLOCK+1	00000160
JBLOCK=NBLOCK	00000170
NBRNCH=0	00000180
10 CALL GNLE	00000190
IF(JTYP .NE. 5) GO TO 30	00000200
CALL STSRCH	00000210
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)	00000220
IF(NBRNCH .EQ. 0) GO TO 15	00000230
DO 12 I=1,NBRNCH	00000240
IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 17	00000250
12 CONTINUE	00000260
15 NBLOCK=NBLOCK+1	00000270
IBLOCK(NBLOCK)=LOC	00000280
NBRNCH=NBRNCH+1	00000290
17 IF(NEXT(JPTR) .EQ. COMMA) GO TO 10	00000300
IF(A(JPTR-1) .NE. RPAR) GO TO 30	00000310
IF(NEXT(JPTR) .NE. COMMA) GO TO 30	00000320
CALL GNLE	00000330
IF(JTYP .NE. 2) GO TO 30	00000340
CALL SEARCH	00000350
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000360
IF(ISRCH(1) .EQ. 1) GO TO 20	00000370
IDTYP=1	00000380
CALL STORE	00000390
LOC=NID	00000400
20 CALL IMPTYP	00000410
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)	00000420
\$ CALL ERROR(39,NXTID(1),NXTID(2),IDM3,IDM4)	00000430
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)	00000440
\$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000450
IF(NEXT(JPTR) .NE. BLANK) GO TO 30	00000460
IBLOCK(JBLOCK)=2000+LOC	00000470
NB=1	00000480
RETURN	00000490
30 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000500
RETURN	00000510
END	00000520



SUBROUTINE DATA	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRC(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
DIMENSION IALPH(4)	00000050
INTEGER A,RPAR,COMMA,SLASH,BLANK,ASTRIK,PLUS	00000060
INTEGER BITPUT,BITGET	00000070
DATA LPAR/1H(/,RPAR/1H(/,COMMA/1H(/,SLASH/1H(/,BLANK/1H /,	00000080
* ASTRIK/1H*/ ,PLUS/1H+/,MINUS/1H-/	00000090
DATA IALPH/1HD,1HA,1MT,1HA/	00000100
DO 5 I=1,4	00000110
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60	00000120
5 CONTINUE	00000130
6 LST1SZ=0	00000140
LST2SZ=0	00000150
8 ISZ=1	00000160
CALL GNLE	00000170
IF(JTYP .NE. 2) GO TO 60	00000180
CALL SEARCH	00000190
IF(ISRC(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000200
IF(ISRC(1) .EQ. 1) GO TO 9	00000210
IDTYP=1	00000220
CALL STORE	00000230
LOC=NID	00000240
9 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)	00000250
% CALL ERROR(30,NXTID(1),NXTID(2),IDM3,IDM4)	00000260
CALL IMPTYP	00000270
IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 0) GO TO 10	00000280
ICOMLC=IDTBL(9,LOC)	00000290
IF(IBLKDT .EQ. 0 .OR. IDTBL(1,ICOMLC) .EQ. BLANK)	00000300
% CALL ERROR(28,NXTID(1),NXTID(2),IDM3,IDM4)	00000310
10 IF(NEXT(JPTR) .NE. LPAR) GO TO 25	00000320
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 90	00000330
NDIM=BITGET(IDTBL(3,LOC),7,6)	00000340
I=0	00000350
15 I=I+1	00000360
CALL GNLE	00000370
IF(JTYP .NE. 5) GO TO 60	00000380
IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)	00000390
IF(N2 .GT. IDTBL(4+I,LOC)) CALL ERROR(18,IDM1,IDM2,IDM3,IDM4)	00000400
IF(NEXT(JPTR) .EQ. COMMA) GO TO 15	00000410
IF(I .NE. NDIM) GO TO 80	00000420
IF(A(JPTR-1) .NE. RPAR) GO TO 80	00000430
GO TO 35	00000440
25 JPTR=JPTR-1	00000450
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 35	00000460
NDIM=BITGET(IDTBL(3,LOC),7,6)	00000470
DO 30 I=1,NDIM	00000480
30 ISZ=ISZ+IDTBL(4+I,LOC)	00000490
IF(BITGET(IDTBL(3,LOC),14,1) .EQ. 1)	00000500
% CALL ERROR(29,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)	00000510
35 LST1SZ=LST1SZ+ISZ	00000520
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,14)	00000530
IF(NEXT(JPTR) .EQ. COMMA) GO TO 8	00000540
IF(A(JPTR-1) .NE. SLASH) GO TO 60	00000550



40 NRPEAT=1	00000560
CALL GNLE	00000570
IF(JTYP .EQ. 3) GO TO 47	00000580
IF(JTYP .NE. 5) GO TO 45	00000590
IF(NEXT(JPTR) .NE. ASTRIK) GO TO 50	00000600
NRPEAT=N2	00000610
CALL GNLE	00000620
45 IF(A(JPTR-1) .NE. PLUS .AND. A(JPTR-1) .NE. MINUS) GO TO 47	00000630
CALL GNLE	00000640
47 KK=NEXT(JPTR)	00000650
50 IF(JTYP .GE. 3 .AND. JTYP .LE. 6) GO TO 55	00000660
IF(JTYP .EQ. 7 .AND. LOGID .EQ. 10) GO TO 55	00000670
IF(JTYP .EQ. 7 .AND. LOGID .EQ. 11) GO TO 55	00000680
GO TO 70	00000690
55 LST2SZ=LST2SZ+NRPEAT	00000700
IF(A(JPTR-1) .EQ. COMMA) GO TO 40	00000710
IF(A(JPTR-1) .NE. SLASH) GO TO 60	00000720
IF(LSTISZ .NE. LST2SZ) CALL ERROR(31, IDM1, IDM2, IDM3, IDM4)	00000730
IF(NEXT(JPTR) .EQ. COMMA) GO TO 6	00000740
IF(A(JPTR-1) .NE. BLANK) GO TO 60	00000750
RETURN	00000760
60 CALL ERROR(7, IDM1, IDM2, IDM3, IDM4)	00000770
RETURN	00000780
70 CALL ERROR(23, IDM1, IDM2, IDM3, IDM4)	00000790
RETURN	00000800
80 CALL ERROR(19, IDM1, IDM2, IDM3, IDM4)	00000810
RETURN	00000820
90 CALL ERROR(13, IDTBL(1, LOC), IDTBL(2, LOC), IDM3, IDM4)	00000830
RETURN	00000840
END	00000850



SURROUTINE DESCRP	00000010
COMMON A(1326),D(500),IDTRL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,	00000030
• LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES	00000040
COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,	00000050
1 DIR,ICOM,ISEP	00000060
DIMENSION FORMT(7)	00000070
INTEGER A,FORMT,DECPT,PEE,EX	00000080
DATA FORMT/1HF,1HE,1HG,1HD,1HI,1HL,1HA/	00000090
DATA DECPT/1H./,PEE/1HP/,EX/1HX/,MINUS/1H-/	00000100
ISCLFC=0	00000110
I-T=0	00000120
IMINUS=0	00000130
IDES=1	00000140
JPTR=IDESST	00000150
IF(NEXT(JPTR) .NE. MINUS) GO TO 5	00000160
IMINUS=1	00000170
GO TO 6	00000180
5 JPTR=IDESST	00000190
6 CONTINUE	00000200
CALL GNLE	00000210
IF(JTYP .EQ. 3) GO TO 80	00000220
IF(JTYP .EQ. 5) GO TO 10	00000230
IF(JTYP .NE. 2) GO TO 15	00000240
IF(NXTID(1) .EQ. PEE .AND. ISCLFC .EQ. 0) GO TO 20	00000250
IF(INT .EQ. 1 .AND. N2 .LT. 1) GO TO 15	00000260
IF(ISCLFC .EQ. 0 .AND. IMINUS .EQ. 1) GO TO 15	00000270
GO TO 25	00000280
10 INT=1	00000290
GO TO 6	00000300
15 IDES=0	00000310
RETURN	00000320
20 IF(INT .EQ. 0) GO TO 15	00000330
ISCLFC=1	00000340
INT=0	00000350
GO TO 6	00000360
25 DO 30 I=1,7	00000370
IF(NXTID(1) .EQ. FORMT(I)) GO TO 45	00000380
30 CONTINUE	00000390
IF(NXTID(1) .NE. EX .OR. ISCLFC .EQ. 1 .OR. INT .EQ. 0) GO TO 15	00000400
GO TO 80	00000410
45 CALL GNLE	00000420
IF(JTYP .NE. 5) GO TO 15	00000430
NWIDTH=N2	00000440
IF(I .LE. 4) GO TO 60	00000450
IF(ISCLFC .EQ. 1 .OR. NWIDTH .LT. 1) GO TO 15	00000460
IF(I .EQ. 7 .AND. NWIDTH .GT. 4) GO TO 15	00000470
IDESND=JPTR-1	00000480
RETURN	00000490
60 IF(NEXT(JPTR) .NE. DECPT) GO TO 15	00000500
IF(NWIDTH .LT. 2) GO TO 15	00000510
CALL GNLE	00000520
IF(JTYP .NE. 5) GO TO 15	00000530
NDCPLS=N2	00000540
IDESND=JPTR-1	00000550
IF(I .EQ. 1) GO TO 65	00000560
IF(NWIDTH .LT. (NDCPLS-6)) GO TO 15	00000570
RETURN	00000580
65 IF(NWIDTH .LT. NDCPLS) GO TO 15	00000590
RETURN	00000600
80 IDESND=JPTR-1	00000610
RETURN	00000620
END	00000630



```

SUBROUTINE DIMEN                                00000210
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISPC(3), 00000220
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NJO, 00000230
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES 00000240
DIMENSION IALPH(9),IDIM(3) 00000250
INTEGER A,D,RPAR,COMMA 00000260
INTEGER BITPUT,BITGET,COMLOC 00000270
DATA IALPH/1HD,1HI,1HM,1HE,1HN,1HS,1HI,1HO,1HN/ 00000280
DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/ 00000290
DO 10 I=1,9 00000300
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 110 00000310
10 CONTINUE 00000320
12 CALL GNLE 00000330
IF(JTYP .NE. 2) GO TO 110 00000340
CALL SEARCH 00000350
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4) 00000360
IF(ISRCH(1) .EQ. 1) GO TO 5 00000370
IDTYP=1 00000380
CALL STORE 00000390
LOC=NID 00000400
5 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) 00000410
$ CALL ERROR(11,NXTID(1),NXTID(2),IDM3,IDM4) 00000420
CALL IMPTYP 00000430
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1) 00000440
IE=LOC 00000450
IF(NEXT(JPTR) .NE. LPAR) GO TO 110 00000460
INCR=1 00000470
I=0 00000480
15 I=I+1 00000490
CALL GNLE 00000500
IF(JTYP .NE. 5) GO TO 13 00000510
IDIM(I)=N2 00000520
IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4) 00000530
IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4) 00000540
INCR=INCR+N2 00000550
GO TO 14 00000560
13 IF(JTYP .NE. 2) GO TO 110 00000570
IDTYP=1 00000580
CALL SEARCH 00000590
IF(ISRCH(2) .NE. 0) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4) 00000600
IF(ISRCH(1) .EQ. 1) GOTO 25 00000610
IDTYP=1 00000620
CALL STORE 00000630
LOC=NID 00000640
25 IF(BITGET(IDTBL(3,LOC),12,1) .NE. 1) 00000650
$ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4) 00000660
IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 120 00000670
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,13) 00000680
CALL IMPTYP 00000690
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) 00000700
$ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4) 00000710
IDIM(I)=2**17+LOC 00000720
14 IF(NEXT(JPTR) .EQ. COMMA) GO TO 15 00000730
IF(A(JPTR-1) .NE. RPAR) GO TO 110 00000740
LOC=IE 00000750

```



IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)	00000560
IF(I.GT. 3) GO TO 110	00000570
DO 30 J=1,I	00000580
30 IDTBL(4+J,LOC)=IDIM(J)	00000590
IF(BITGET(IDTBL(3,LOC),16,1).NE. 1) GO TO 50	00000600
COMLOC=IDTBL(9,LOC)	00000610
IT=1	00000620
ITP=BITGET(IDTBL(3,LOC),10,3)	00000630
IF(ITP.EQ. 2.OR. ITP.EQ. 3) IT=2	00000640
IDTBL(5,COMLOC)=IDTBL(5,COMLOC)+IT*(INCR-1)	00000650
50 CONTINUE	00000660
IF(NEXT(JPTR).EQ. COMMA) GO TO 12	00000670
IF(JPTR.GT. N) RETURN	00000680
110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000690
RETURN	00000700
120 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000710
RETURN	00000720
END	00000730



```

SUBROUTINE DO
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISPC(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NTD,
* LOC,LTYP,ITYP,IBLKDT,MODE,IEHR,IDES
COMMON/LABELS/STATRA(2,200),NLABEL
COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW
COMMON/HASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
DIMENSION PARAM(3)
INTEGER A,BLANK,COMMA,EQUALS,DEE,OH,PARAM,STATRA
INTEGER BITPUT,BITGET
DATA BLANK/1H /,COMMA/1H /,EQUALS/1H =/,DEE/1MD/,OH/1HO/
IF(NEXT(JPTR) .NE. DEE) GO TO 50
IF(NEXT(JPTR) .NE. OH) GO TO 50
CALL GNLE
IF(JTYP .NE. 5) GOTO 50
CALL STSRCH
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,15)
IF(IOVFLW .EQ. 1) GO TO 2
NSTACK=NSTACK+1
IF(NSTACK .GT. 50) GO TO 1
ISTACK(1,NSTACK)=LOC
ISTACK(2,NSTACK)=0
ISTACK(3,NSTACK)=ILOOP
ILOOP=NSTACK
GO TO 2
1 IOVFLW=1
WRITE(6,60)
60 FORMAT(///5X,50H DO STACK OVERFLOW - DO LOOP PROCESSING TERMINATED
*)
2 CALL GNLE
IF(JTYP .NE. 2) GO TO 50
CALL SEARCH
IF(ISPC(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(1) .EQ. 1) GO TO 5
IDTYP=1
CALL STORE
LOC=NID
5 CALL IMPTYP
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)
$ CALL ERROR(40,NXTID(1),NXTID(2),IDM3,IDM4)
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)
$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)
IF(NEXT(JPTR) .NE. EQUALS) GO TO 50
IF(IOVFLW .EQ. 1) GO TO 8
NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=3000+LOC
ISTACK(4,NSTACK)=LOC
8 PARAM(3)=1
DO 30 I=1,3
CALL GNLE
IF(JTYP .NE. 5) GO TO 10
PARAM(I)=N2
IF(N2 .LE. 0) CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)
GO TO 20

```

```

00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```



10 IF(JTYP .NE. 2) GO TO 50	00000560
CALL SEARCH	00000570
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000580
IF(ISRCH(1) .EQ. 1) GO TO 15	00000590
IDTYP=1	00000600
CALL STORE	00000610
LOC=NID	00000620
15 CALL IMPTYP	00000630
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)	00000640
\$ CALL ERROR(40,NXTID(1),NXTID(2),IDM3,IDM4)	00000650
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1)	00000660
\$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000670
NBLOCK=NBLOCK+1	00000680
IBLOCK(NBLOCK)=7000+LOC	00000690
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ILOOP,32)	00000700
PARAM(1)=0	00000710
20 IF(I .EQ. 3) GO TO 30	00000720
IF(I .EQ. 1) GO TO 25	00000730
IF(NEXT(JPTR) .EQ. BLANK) GO TO 35	00000740
JPTR=JPTR-1	00000750
25 IF(NEXT(JPTR) .NE. COMMA) GO TO 50	00000760
30 CONTINUE	00000770
IF(NEXT(JPTR) .NE. BLANK) GO TO 50	00000780
35 IF(PARAM(1) .EQ. 0 .OR. PARAM(2) .EQ. 0) GO TO 40	00000790
IF(PARAM(2) .LT. PARAM(1)) CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)	00000800
IF(PARAM(3) .EQ. 0) GO TO 40	00000810
IF((PARAM(2)+PARAM(3)-1) .GT. 2**17-1)	00000820
\$ CALL ERROR(41,IDM1,IDM2,IDM3,IDM4)	00000830
40 NBLOCK=NBLOCK+1	00000840
IBLOCK(NBLOCK)=998	00000850
NBRNCH=1	00000860
NB=1	00000870
RETURN	00000880
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000890
RETURN	00000900
END	00000910



```

SUBROUTINE EQUIV
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,INTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
DIMENSION IALPH(11),IDIM(3)
INTEGER BITPUT,BITGET
INTEGER A,RPAR,COMMA,BOFFST,HLANK
DATA IALPH/1HE,1HQ,1HU,1HI,1HV,1HA,1HL,1HE,1HN,1HC,1HE/
DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H/,BLANK/1H /
DO 5 I=1,11
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 130
5 CONTINUE
8 IF(NEXT(JPTR) .NE. LPAR) GO TO 130
LSTLOC=0
BOFFST=0
J=0
120 J=J+1
CALL GNLE
ILOC=1
IF(JTYP .NE. 2) GO TO 130
CALL SEARCH
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)
IF(ISRCH(1) .EQ. 1) GO TO 9
IDTYP=1
CALL STORE
LOC=NID
9 CALL IMPTYP
IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1)
$ CALL ERROR(20,NXTID(1),NXTID(2),IDM3,IDM4)
IF(NEXT(JPTR) .NE. LPAR) GO TO 30
IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) GO TO 150
NDIM=BITGET(IDTBL(3,LOC),7,6)
DO 10 I=1,NDIM
CALL GNLE
IF(JTYP .NE. 5) GO TO 130
IDIM(I)=N2
IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)
IF(N2 .GT. IDTBL(4+1,LOC)) CALL ERROR(18,IDM1,IDM2,IDM3,IDM4)
IF(NEXT(JPTR) .EQ. COMMA) GO TO 10
IF(A(JPTR-1) .NE. RPAR) GO TO 130
GO TO 15
10 CONTINUE
GO TO 140
15 NDIM=I
ILOC=IDIM(1)
IF(NDIM .EQ. 1) GO TO 25
ILOC=ILOC+(IDIM(2)-1)*IDTBL(5,LOC)
IF(NDIM .EQ. 2) GO TO 25
ILOC=ILOC+(IDIM(3)-1)*IDTBL(5,LOC)*IDTBL(6,LOC)
25 IT=BITGET(IDTBL(3,LOC),10,3)
IF(IT .EQ. 2 .OR. IT .EQ. 3) ILOC=2*ILOC
IOFFST=1-ILOC-BOFFST
GO TO 45
30 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)
$ CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)

```



IOFFST=ROFFST	00000560
JPTR=JPTR-1	00000570
45 IF (BITGET(IDTBL(3,LOC),17,1) .EQ. 1) GO TO 57	00000580
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,17)	00000590
IF (LSTLOC .EQ. 0) GO TO 50	00000600
IDTBL(10,LSTLOC)=LOC	00000610
GO TO 55	00000620
50 IFSTLC=LOC	00000630
55 IDTBL(11,LOC)=IOFFST	00000640
LSTLOC=LOC	00000650
IF (J .NE. 1) GO TO 100	00000660
GO TO 98	00000670
57 LOC3=LOC	00000680
58 LOC3=IDTBL(10,LOC3)	00000690
IF (LOC3 .EQ. 0) GO TO 59	00000700
IF (LOC3 .EQ. LOC) GO TO 60	00000710
GO TO 58	00000720
59 JLOC=ILOC+IDTBL(11,LOC)	00000730
IF (JLOC .NE. IDIS) CALL ERROR(20,IDTBL(1,LOC),IDTBL(2,LOC),	00000740
\$ IDM3,IUM4)	00000750
GO TO 100	00000760
60 IF (LSTLOC .NE. 0) GO TO 63	00000770
IFSTLC=LOC	00000780
GO TO 65	00000790
63 IDTBL(10,LSTLOC)=LOC	00000800
65 LOC2=LOC	00000810
70 NXTLOC=IDTBL(10,LOC2)	00000820
IF (NXTLOC .EQ. LOC) GO TO 75	00000830
LOC2=NXTLOC	00000840
GO TO 70	00000850
75 IDTBL(10,LOC2)=0	00000860
LSTLOC=LOC2	00000870
IOFFDF=IOFFST-IDTBL(11,LOC)	00000880
IF (IOFFDF) 80,9H,90	00000890
80 LOC2=LOC	00000900
85 IDTBL(11,LOC2)=IDTBL(11,LOC2)+IOFFDF	00000910
LOC2=IDTBL(10,LOC2)	00000920
IF (LOC2 .EQ. 0) GO TO 98	00000930
GO TO 85	00000940
90 LOC2=IFSTLC	00000950
95 IF (LOC2 .EQ. LOC) GO TO 97	00000960
IDTBL(11,LOC2)=IDTBL(11,LOC2)-IOFFDF	00000970
LOC2=IDTBL(10,LOC2)	00000980
GO TO 95	00000990
97 ROFFST=ROFFST-IOFFDF	00001000
98 IDIS=ILOC+IDTBL(11,LOC)	00001010
100 IF (NEXT(JPTR) .EQ. COMMA) GO TO 120	00001020
IF (A(JPTR-1) .NE. RPAR) GO TO 130	00001030
IF (J .EQ. 1) GO TO 130	00001040
IDTBL(10,LSTLOC)=IFSTLC	00001050
JK=0	00001060
LOC3=IFSTLC	00001070
I=0	00001080
110 I=I+1	00001090
LOC3=IDTBL(10,LOC3)	00001100



```

IF (BITGET(IDTBL(3,LOC3),16,1) .EQ. 0) GO TO 105
JK=JK+1
IF (JK .GT. 1) CALL ERROR(21,IDTBL(1,LOC3),IDTBL(2,LOC3),
& IDTBL(1,LOC1),IDTBL(2,LOC1))
LOC1=LOC3
LOC=LOC3
CALL COMEXT
105 IF (LOC3 .NE. IFSTLC) GO TO 110
IF (NEXT(JPTR) .EQ. BLANK) RETURN
IF (A(JPTR-1) .EQ. COMMA) GO TO 8
130 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)
RETURN
140 CALL ERROR(19,IDM1,IDM2,IDM3,IDM4)
RETURN
150 CALL ERROR(13,NXTID(1),NXTID(2),IDM3,IDM4)
RETURN
END

```

```

00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270

```



```

SUBROUTINE ERROR(IEERROR, IDM1, IDM2, IDM3, IDM4)
COMMON A(1326), D(500), IDTBL(11,500), INITID(3), LASTID(3), ISRCH(3),
* NXTID(2), JPTR, N, M, JTYPE, LSTART, N2, IFNCNM, LOGIO, IOTYP, NID,
$LOC, LTYPE, ITYP, IBLKDT, MODE, IERR, IDES
COMMON/FLOW/IFL, IRP
WRITE(6,1)
1 FORMAT(1X,100H.....)
1.....)
GO TO (5,15,25,35,45,55,65,75,85,95,105,115,125,135,145,155,165,
A175,185,195,205,215,225,235,245,255,265,275,285,295,305,315,325,
* 335,345,355,365,375,385,395,405,415,425,435,445,455,465,475,485,
* 495,505,515,525,535,545,555,565,575,585,595,605,615,625,635,645,
$ 655,665,675,685,695,705,715,725,735,745,755,765,775,785,795,805,
$ 815,825,835,845,855,865,875,885,895,905,915,925,935,945), IEERROR
5 WRITE(6, 10)
10 FORMAT(6X,26H THIS STATEMENT IS ILLEGAL)
GO TO 1000
15 WRITE(6, 20)
20 FORMAT(6X,31H THIS STATEMENT IS OUT OF ORDER)
GO TO 1000
25 WRITE(6, 30)
30 FORMAT(6X,39H VALUE OF INTEGER CONSTANT IS TOO LARGE)
GO TO 1000
35 WRITE(6, 40)
40 FORMAT(6X,28H TOO MANY CONTINUATION CARDS)
GO TO 1000
45 WRITE(6, 50)
50 FORMAT(6X,30H HOLLERITH STRING IS TOO LARGE)
GO TO 1000
55 WRITE(6, 60)
60 FORMAT(6X,26H VARIABLE NAME IS TOO LONG)
GO TO 1000
65 WRITE(6, 70)
70 FORMAT(6X,31H SYNTAX ERROR IN THIS STATEMENT)
IF (ITYP .LE. 18 .AND. IFL .GT. 0) IFL=-1
GO TO 1000
75 WRITE(6, 80)
80 FORMAT(6X,46H ARRAY DIMENSION IS OUTSIDE OF ALLOWABLE RANGE)
GO TO 1000
85 WRITE(6, 90)
90 FORMAT(6X,45H ILLEGAL VARIABLE DIMENSION IN THIS STATEMENT)
GO TO 1000
95 WRITE(6,100) IDM1, IDM2
100 FORMAT(6X,33H THE FUNCTION OR SUBROUTINE NAME ,A4,A2,1RH IS USED I
$LLLEGALLY)
GO TO 1000
105 WRITE(6,110) IDM1, IDM2
110 FORMAT(6X,14H THE VARIABLE ,A4,A2,32H HAS BEEN PREVIOUSLY DIMENSIO
$NED)
GO TO 1000
115 WRITE(6,120) IDM1, IDM2
120 FORMAT(6X,14H THE VARIABLE ,A4,A2,26H HAS BEEN PREVIOUSLY TYPED)
GO TO 1000
125 WRITE(6,130) IDM1, IDM2
130 FORMAT(6X,14H THE VARIABLE ,A4,A2,38H IS ILLEGALLY FOLLOWED BY A L

```



LEFT PAREN)	00000560
GO TO 1000	00000570
135 WRITE(6,140) ID1, ID2	00000580
140 FORMAT(6X,26H THE DIMENSIONED VARIABLE ,A4,A2,18H IS USED ILLEGAL	00000590
SY)	00000600
GO TO 1000	00000610
145 WRITE(6,150) ID1	00000620
150 FORMAT(6X,18H STATEMENT NUMBER ,I5,15H IS NOT DEFINED)	00000630
IF (IFL .GT. 0) IFL=-1	00000640
GO TO 1000	00000650
155 WRITE(6,160) ID1	00000660
160 FORMAT(6X,18H STATEMENT NUMBER ,I5,18H IS NOT REFERENCED)	00000670
GO TO 1000	00000680
165 WRITE(6,170) ID1, ID2	00000690
170 FORMAT(6X,18H ILLEGAL VARIABLE ,A4,A2,10H IN COMMON)	00000700
GO TO 1000	00000710
175 WRITE(6,180)	00000720
180 FORMAT(6X,43H VALUE OF ARRAY SUBSCRIPT EXCEEDS DIMENSION)	00000730
GO TO 1000	00000740
185 WRITE(6,190)	00000750
190 FORMAT(6X,25H ERROR IN ARRAY SUBSCRIPT)	00000760
GO TO 1000	00000770
195 WRITE(6,200) ID1, ID2	00000780
200 FORMAT(6X,18H ILLEGAL VARIABLE ,A4,A2,16H IS EQUIVALENCED)	00000790
GO TO 1000	00000800
205 WRITE(6,210) ID1, ID2, ID3, ID4	00000810
210 FORMAT(6X,22H THE COMMON VARIABLES ,A4,A2,5H AND ,A4,A2,17H ARE EQ	00000820
SUIVALENCED)	00000830
GO TO 1000	00000840
215 WRITE(6,220)	00000850
220 FORMAT(6X,19H ILLEGAL I/O DEVICE)	00000860
GO TO 1000	00000870
225 WRITE(6,230)	00000880
230 FORMAT(6X,37H ILLEGAL CHARACTER IN THIS EXPRESSION)	00000890
GO TO 1000	00000900
235 WRITE(6,240) ID1, ID2	00000910
240 FORMAT(6X,25H ILLEGAL SUBROUTINE NAME ,A4,A2)	00000920
GO TO 1000	00000930
245 WRITE(6,250)	00000940
250 FORMAT(6X,50H SUBROUTINE TABLE OVERFLOW - PROCESSING TERMINATED)	00000950
GO TO 1000	00000960
255 WRITE(6,260) ID1, ID2	00000970
260 FORMAT(6X,77H INCORRECT NUMBER OF ARGUMENTS IN CALLING SEQUENCE OF	00000980
FUNCTION OR SUBROUTINE ,A4,A2)	00000990
GO TO 1000	00001000
265 WRITE(6,270)	00001010
270 FORMAT(6X,19H ILLEGAL ASSIGNMENT)	00001020
GO TO 1000	00001030
275 WRITE(6,280) ID1, ID2	00001040
280 FORMAT(6X,14H THE VARIABLE ,A4,A2,42H APPEARS IN A DATA STATEMENT	00001050
&AND IN COMMON)	00001060
GO TO 1000	00001070
285 WRITE(6,290) ID1, ID2	00001080
290 FORMAT(6X,14H THE VARIABLE ,A4,A2,44H HAS PREVIOUSLY APPEARED IN A	00001090
DATA STATEMENT)	00001100



GO TO 1000	00001110
295 WRITE(6,300) IDM1,IDM2	00001120
300 FORMAT(6X,22H THE FORMAL PARAMETER ,A4,A2,31H APPEARS IN THIS DATA	00001130
\$ STATEMENT)	00001140
GO TO 1000	00001150
305 WRITE(6,310)	00001160
310 FORMAT(6X,24H LIST SIZES DO NOT MATCH)	00001170
GO TO 1000	00001180
315 WRITE(6,320)	00001190
320 FORMAT(6X,24H ILLEGAL STATEMENT LABEL)	00001200
IF(IFL .GT. 0) IFL=-1	00001210
GO TO 1000	00001220
325 WRITE(6,330)	00001230
330 FORMAT(6X,26H DUPLICATE STATEMENT LABEL)	00001240
IF(IFL .GT. 0) IFL=-1	00001250
GO TO 1000	00001260
335 WRITE(6,340)	00001270
340 FORMAT(6X,34H THIS STATEMENT CAN NOT BE REACHED)	00001280
GO TO 1000	00001290
345 WRITE(6,350)	00001300
350 FORMAT(6X,31H DO LOOPS ARE IMPROPERLY NESTED)	00001310
GO TO 1000	00001320
355 WRITE(6,360)	00001330
360 FORMAT(6X,32H FORMAT STATEMENT IS NOT LABELED)	00001340
GO TO 1000	00001350
365 WRITE(6,370)	00001360
370 FORMAT(6X,20H ILLEGAL DO TERMINAL)	00001370
GO TO 1000	00001380
375 WRITE(6,380)	00001390
380 FORMAT(6X,37H LAST EXECUTABLE STATEMENT IS ILLEGAL)	00001400
IF(IFL .GT. 0) IFL=-1	00001410
GO TO 1000	00001420
385 WRITE(6,390) IDM1,IDM2	00001430
390 FORMAT(6X,24H THE VARIABLE REFERENCE ,A4,A2,18H IS NOT AN INTEGER)	00001440
GO TO 1000	00001450
395 WRITE(6,400) IDM1,IDM2	00001460
400 FORMAT(6X,27H THE DO PARAMETER OR INDEX ,A4,A2,18H IS NOT AN INTEG	00001470
\$ER)	00001480
GO TO 1000	00001490
405 WRITE(6,410)	00001500
410 FORMAT(6X,52H VALUE OF DO PARAMETER IS OUTSIDE OF ALLOWABLE RANGE)	00001510
GO TO 1000	00001520
415 WRITE(6,420)	00001530
420 FORMAT(6X,32H COMPLEX EXPRESSIONS ARE ILLEGAL)	00001540
GO TO 1000	00001550
425 WRITE(6,430)	00001560
430 FORMAT(6X,24H ILLEGAL VARIABLE FORMAT)	00001570
GO TO 1000	00001580
435 WRITE(6,440)	00001590
440 FORMAT(6X,39H THIS STATEMENT SHOULD HAVE AN I/O LIST)	00001600
GO TO 1000	00001610
445 WRITE(6,450)	00001620
450 FORMAT(6X,50H STATEMENT FOLLOWING LOGICAL EXPRESSION IS ILLEGAL)	00001630
GO TO 1000	00001640
455 WRITE(6,460)	00001650



460	FORMAT(6X,44H REAL NUMBER LIES OUTSIDE OF ALLOWABLE RANGE)	00001660
	GO TO 1000	00001670
465	WRITE(6,470)	00001680
470	FORMAT(6X,42H THIS EQUIVALENCE STATEMENT EXTENDS COMMON)	00001690
	GO TO 1000	00001700
475	WRITE(6,480)	00001710
480	FORMAT(6X,40H ILLEGAL VARIABLE IN COMMON BLOCK SESCOM)	00001720
	GO TO 1000	00001730
485	WRITE(6,490) IDM1,IDM2	00001740
490	FORMAT(6X,12H SUBPROGRAM ,A4,A2,19H HAS INCORRECT TYPE)	00001750
	GO TO 1000	00001760
495	WRITE(6,500) IDM1	00001770
500	FORMAT(6X,23H WARNING - ARGUMENT NO.,I3,34H MAY HAVE INCORRECT DIMENSIONALITY)	00001780
	GO TO 1000	00001790
505	WRITE(6,510) IDM1	00001800
510	FORMAT(6X,13H ARGUMENT NO.,I3,19H HAS INCORRECT TYPE)	00001810
	GO TO 1000	00001820
515	WRITE(6,520)	00001830
520	FORMAT(6X,49H WARNING - THIS MODULE IS NOT IN THE SESCOMP LIST)	00001840
	GO TO 1000	00001850
525	WRITE(6,530) IDM1,IDM2	00001860
530	FORMAT(6X,14H THE VARIABLE ,A4,A2,29H PREVIOUSLY APPEARS IN COMMON BLOCK \$)	00001870
	GO TO 1000	00001880
535	WRITE(6,540) IDM1	00001890
540	FORMAT(6X,13H ARGUMENT NO.,I3,11H IS INVALID)	00001900
	GO TO 1000	00001910
545	WRITE(6,550) IDM1	00001920
550	FORMAT(6X,13H ARGUMENT NO.,I3,29H IS DESIGNATED LOGICAL OUTPUT)	00001930
	GO TO 1000	00001940
555	WRITE(6,560) IDM1,IDM2	00001950
560	FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A4,A2)	00001960
	GO TO 1000	00001970
565	WRITE(6,570) IDM1,IDM2	00001980
570	FORMAT(6X,41H WARNING - VARIABLE TYPE IN COMMON BLOCK ,A4,A2,41H DOES NOT AGREE WITH INTERFACE DEFINITION)	00001990
	GO TO 1000	00020000
575	WRITE(6,580) IDM1,IDM2	00002010
580	FORMAT(6X,14H COMMON BLOCK ,A4,A2,19H HAS INCORRECT SIZE)	00002020
	GO TO 1000	00002030
585	WRITE(6,590)	00002040
590	FORMAT(6X,58H EXTERNAL REFERENCE TABLE OVERFLOW - PROCESSING TERMINATED)	00002050
	GO TO 1000	00002060
595	WRITE(6,600)	00002070
600	FORMAT(6X,52H COMMON BLOCK TABLE OVERFLOW - PROCESSING TERMINATED)	00002080
	GO TO 1000	00002090
605	WRITE(6,610) IDM1,IDM2	00002100
610	FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A4,A2)	00002110
	GO TO 1000	00002120
615	WRITE(6,620) IDM1,IDM2	00002130
620	FORMAT(6X,14H COMMON BLOCK ,A4,A2,27H IS NOT IN THE SESCOMP LIST)	00002140
	GO TO 1000	00002150
625	WRITE(6,630) IDM1,IDM2	00002160
		00002170
		00002180
		00002190
		00002200



```

630 FORMAT(6X,25H CATEGORY 2 COMMON BLOCK ,A4,A2,23H IS NOT GROUPED BY00002210
$ TYPE) 00002220
GO TO 1000 00002230
635 WRITE(6,640) IDM1,IDM2,IDM3,IDM4 00002240
640 FORMAT(6X,38H DOUBLE PRECISION OR COMPLEX VARIABLE ,A4,A2,56H DOES00002250
$ NOT BEGIN ON AN EVEN LOCATION WITHIN COMMON BLOCK ,A4,A2) 00002260
GO TO 1000 00002270
645 WRITE(6,650) IDM1,IDM2 00002280
650 FORMAT(6X,26H VARIABLE IN COMMON BLOCK ,A4,A2,16H IS OUT OF ORDER)00002290
GO TO 1000 00002300
655 WRITE(6,660) 00002310
660 FORMAT(6X,56H THE COMMON BLOCK SESCOM DOES NOT APPEAR IN THIS PROG00002320
$RAM) 00002330
GO TO 1000 00002340
665 WRITE(6,670) IDM1,IDM2 00002350
670 FORMAT(6X,14H THE DO INDEX ,A4,A2,13H IS REDEFINED) 00002360
RETURN 00002370
675 WRITE(6,680) IDM1,IDM2 00002380
680 FORMAT(6X,24H THE VARIABLE DIMENSION ,A4,A2,13H IS REDEFINED) 00002390
RETURN 00002400
685 WRITE(6,690) IDM1,IDM2 00002410
690 FORMAT(6X,23H THE ASSIGNED VARIABLE ,A4,A2,24H IS ILLEGALLY REFERE00002420
$NCED) 00002430
RETURN 00002440
695 WRITE(6,700) IDM1,IDM2 00002450
700 FORMAT(6X,14H THE VARIABLE ,A4,A2,30H IS REFERENCED BUT NOT DEFINE00002460
$D) 00002470
RETURN 00002480
705 WRITE(6,710) IDM1,IDM2 00002490
710 FORMAT(6X,14H THE VARIABLE ,A4,A2,45H IS REFERENCED ILLEGALLY BY A00002500
$N ASSIGNED GO TO) 00002510
RETURN 00002520
715 WRITE(6,720) IDM1,IDM2 00002530
720 FORMAT(6X,18H THE DO PARAMETER ,A4,A2,13H IS REDEFINED) 00002540
RETURN 00002550
725 WRITE(6,730) 00002560
730 FORMAT(6X,49H THIS MODULE CONTAINS NO CATEGORY 2 COMMON BLOCKS) 00002570
GO TO 1000 00002580
735 WRITE(6,740) IDM1,IDM2 00002590
740 FORMAT(6X,24H THE ANSI FUNCTION NAME ,A4,A2,27H IS MISUSED IN THIS00002600
$ PROGRAM) 00002610
GO TO 1000 00002620
745 WRITE(6,750) IDM1,IDM2 00002630
750 FORMAT(6X,14H THE VARIABLE ,A4,A2,60H APPEARS IN A CATEGORY 2 OR 300002640
$ COMMON BLOCK BUT IS NEVER USED) 00002650
GO TO 1000 00002660
755 WRITE(6,760) 00002670
760 FORMAT(6X,75H ARRAY SUBSCRIPT OR IMPLIED DO PARAMETER MAY LIE OUTS00002680
$IDE OF ALLOWABLE RANGE) 00002690
GO TO 1000 00002700
765 WRITE(6,770) IDM1,IDM2,IDM3,IDM4 00002710
770 FORMAT(6X,22H MIXED MODE COMBINING ,A4,A2,6H WITH ,A4,A2) 00002720
GO TO 1000 00002730
775 WRITE(6,780) IDM1 00002740
780 FORMAT(6X,33H INCORRECT EXPONENT AT CHAR. NO. ,I3) 00002750

```



GO TO 1000	00002760
785 WRITE(6,790) IDM1	00002770
790 FORMAT(6X,47H VAR-CONST CONFUSION IN SUBSCRIPT AT CHAR. NO. ,I3)	00002780
GO TO 1000	00002790
795 WRITE(6,800) IDM1,IDM2,IDM3	00002800
800 FORMAT(6X,40H SUBSCRIPT CONSTANT OR VARIABLE OF TYPE ,A4,A2,14H AT	00002810
CHAR. NO. ,I3)	00002820
GO TO 1000	00002830
805 WRITE(6,810) IDM1	00002840
810 FORMAT(6X,52H TOO MANY SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,	00002850
I3)	00002860
GO TO 1000	00002870
815 WRITE(6,820) IDM1	00002880
820 FORMAT(6X,51H TOO FEW SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,	00002890
I3)	00002900
GO TO 1000	00002910
825 WRITE(6,830) IDM1	00002920
830 FORMAT(6X,52H ILLEGAL TYPE IN RELATIONAL EXPRESSION AT CHAR. NO. ,	00002930
I3)	00002940
GO TO 1000	00002950
835 WRITE(6,840) IDM1	00002960
840 FORMAT(6X,40H TOO MANY ARGUMENTS IN CALLING SEQUENCE ,I3)	00002970
GO TO 1000	00002980
845 WRITE(6,850)	00002990
850 FORMAT(6X,41H TOO MANY FUNCTION REFS IN THIS STATEMENT)	00003000
GO TO 1000	00003010
855 WRITE(6,860) IDM1,IDM2	00003020
860 FORMAT(6X,26H INVALID FORMAL PARAMETER ,A4,A2)	00003030
GO TO 1000	00003040
865 WRITE(6,870) IDM1,IDM2	00003050
870 FORMAT(6X,19H THE FUNCTION NAME ,A4,A2,33H MAY HAVE BEEN PREVIOUSL	00003060
Y MISUSED)	00003070
GO TO 1000	00003080
875 WRITE(6,880) IDM1	00003090
880 FORMAT(6X,39H ILLEGAL FIELD DESCRIPTOR AT CHAR. NO. ,I4)	00003100
GO TO 1000	00003110
885 WRITE(6,890)	00003120
890 FORMAT(6X,38H TOO MANY FUNCTION DEFINING STATEMENTS)	00003130
GO TO 1000	00003140
895 WRITE(6,900)	00003150
900 FORMAT(6X,71H TOO MANY EXTERNAL REFERENCES IN THIS STATEMENT - PRO	00003160
CESSING TERMINATED)	00003170
GO TO 1000	00003180
905 WRITE(6,910)	00003190
910 FORMAT(6X,46H STATEMENT IS TOO LONG - PROCESSING TERMINATED)	00003200
GO TO 1000	00003210
915 WRITE(6,920)	00003220
920 FORMAT(6X,46H SESCOPE LIST OVERFLOW - PROCESSING TERMINATED)	00003230
GO TO 1000	00003240
925 WRITE(6,930)	00003250
930 FORMAT(6X,63H OVERFLOW OF INTERFACE DEFINITION TABLE - PROCESSING	00003260
TERMINATED)	00003270
GO TO 1000	00003280
935 WRITE(6,940)	00003290
940 FORMAT(6X,32H TOO MANY EQUIVALENCED VARIABLES)	00003300
GO TO 1000	00003310
945 WRITE(6,950)	00003320
950 FORMAT(6X,61H TOO MANY VARIABLES IN THIS STATEMENT - PROCESSING TE	00003330
RMINATED)	00003340
1000 WRITE(6,1)	00003350
RETURN	00003360
END	00003370



SUBROUTINE EXPR	00000010
COMMON A(1326),D(500),IDTRL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC	00000050
COMMON/STRING/NTYPE,NSTR,STR(500)	00000060
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)	00000070
COMMON/HASHLK/INLOCK(2500),NHLOCK,NB,NBRNCH	00000080
INTEGER FNCLOC,OPRA(6),HITPII,HITGET	00000090
INTEGER D,ASTRIK,EQUALS,STR,A,RPAR	00000100
DATA OPRA/1H+,1H-,1H/,1H(,1H),1H,/	00000110
DATA ASTRIK/1H*,1H=,1H=/,RPAR/1H)/,LPAR/1H(/	00000120
LP=0	00000130
NFUNC=0	00000140
K=0	00000150
IEXPST=NBLOCK+1	00000160
MARGS=0	00000170
200 K=K+1	00000180
CALL GNLE	00000190
IF(JTYP .EQ. 0) RETURN	00000200
IF(JTYP .NE. 1) GO TO 20	00000210
IF(LTYP .EQ. 9 .OR. ITYP .EQ. 6) GO TO 2	00000220
IF(ITYP .EQ. 1 .OR. ITYP .EQ. 35) GO TO 1	00000230
GO TO 5	00000240
1 IF(D(1) .EQ. EQUALS) RETURN	00000250
GO TO 5	00000260
2 IF(D(1) .EQ. RPAR .AND. LP .EQ. 1) RETURN	00000270
5 DO 10 I=1,6	00000280
IF(D(1) .NE. OPRA(I)) GO TO 10	00000290
STR(K)=-I	00000300
IF(I .EQ. 4) GO TO 6	00000310
IF(I .EQ. 5) GO TO 7	00000320
GO TO 100	00000330
6 LP=LP+1	00000340
GO TO 100	00000350
7 LP=LP-1	00000360
GO TO 100	00000370
10 CONTINUE	00000380
IF(D(1) .NE. EQUALS) GO TO 12	00000390
STR(K)=-18	00000400
GO TO 100	00000410
12 IF(D(1) .NE. ASTRIK) GO TO 110	00000420
IF(D(2) .EQ. ASTRIK .AND. M .GT. 1) GO TO 15	00000430
STR(K)=-7	00000440
GO TO 100	00000450
15 STR(K)=-8	00000460
GO TO 100	00000470
20 IF(JTYP .NE. 7) GO TO 30	00000480
IF(LOGID .GT. 9) GO TO 25	00000490
STR(K)=- (LOGID+8)	00000500
GO TO 100	00000510
25 STR(K)=LSTART+440000+M*1000000	00000520
GO TO 100	00000530
30 IF(JTYP .NE. 4) GO TO 40	00000540
IF(IDES .EQ. 0) GO TO 35	00000550



STR(K)=LSTART+420000*M*1000000	00000560
GO TO 100	00000570
35 CONTINUE	00000580
STR(K)=LSTART+400000*M*1000000	00000590
GO TO 100	00000600
40 IF(JTYP .NE. 6) GO TO 50	00000610
STR(K)=LSTART+410000*M*1000000	00000620
GO TO 100	00000630
50 IF(JTYP .NE. 5) GO TO 55	00000640
STR(K)=LSTART+430000*M*1000000	00000650
GO TO 100	00000660
55 IF(JTYP .NE. 3) GO TO 60	00000662
STR(K)=LSTART+450000*M*1000000	00000664
IF(ITYP .NE. 8) CALL ERROR(23, IDM1, IDM2, IDM3, IDM4)	00000666
GO TO 100	00000668
60 IF(JTYP .NE. 2) GO TO 110	00000670
CALL SEARCH	00000680
IBETA=0	00000690
IF(NEXT(JPTR) .NE. LPAR) GO TO 64	00000700
IF(ISRCH(1) .EQ. 0) GO TO 62	00000710
IF(BITGET(IDTHL(3,LOC),1,1) .EQ. 1) GO TO 67	00000720
CALL SWITCH	00000730
IBETA=5	00000740
GO TO 63	00000750
62 IBETA=5	00000760
IF(ISRCH(2) .EQ. 1) GO TO 63	00000770
IDTYP=2	00000780
CALL STORE	00000790
LOC=NID	00000800
DO 70 I=1,NLIST	00000810
IF(ISURLT(1,I) .NE. IDTHL(1,LOC)) GO TO 70	00000820
IF(ISURLT(2,I) .NE. IDTHL(2,LOC)) GO TO 70	00000830
IF(BITGET(ISURLT(3,I),10,4) .NE. 4) GO TO 63	00000840
ITP=BITGET(ISURLT(3,I),13,3)	00000850
IDTBL(3,LOC)=BITPUT(IDTHL(3,LOC),1,11)	00000860
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ITP,10)	00000870
GO TO 63	00000880
70 CONTINUE	00000890
63 NFUNC=NFUNC+1	00000900
IF(NFUNC .GT. 5) GO TO 120	00000905
FNCLOC(NFUNC)=LOC	00000910
GO TO 68	00000920
64 IF(ISRCH(2) .NE. 1) GO TO 65	00000930
IF(LOC .NE. IFNCNM) CALL ERROR(10, NXTID(1), NXTID(2), IDM1, IDM2)	00000940
65 IF(ISRCH(1) .EQ. 1) GO TO 67	00000950
IDTYP=1	00000960
CALL STORE	00000970
LOC=NID	00000980
GO TO 68	00000990
67 IBETA=BITGET(IDTBL(3,LOC),7,6)	00001000
IF(LOC .NE. IFNCNM) GO TO 68	00001010
IBETA=0	00001020
LOC=IDES	00001030
68 CALL IMPTYP	00001040
IALPH=BITGET(IDTBL(3,LOC),10,3)-1	00001050



JPTR=JPTR-1	00001060
STR(K)=LOC+10000*IALPH+100000*IBETA+1000000*M	00001070
100 NSTR=K	00001080
IF(NSTR.GT. 500) GO TO 130	00001085
GO TO 200	00001090
110 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)	00001100
RETURN	00001110
120 CALL ERROR(90,IDM1,IDM2,IDM3,IDM4)	00001112
STOP	00001114
130 CALL ERROR(91,IDM1,IDM2,IDM3,IDM4)	00001116
STOP	00001118
END	00001120

SUBROUTINE EXPRCK	00000010
COMMON A(1326),D(500),IDTRL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP	00000050
DIMENSION IA(5,5)	00000060
INTEGER RHSTYP	00000070
DATA IA/1.0,0.1,0.0,1.0,0.0,1.0,0.0,1.0,1.1,0.1,0.1,1.0,0.0,0.0,1/	00000080
IF(IA(LHSTYP,RHSTYP+1).EQ. 0) CALL ERROR(27,IDM1,IDM2,IDM3,IDM4)	00000090
RETURN	00000100
END	00000110



```

SUBROUTINE FLOWCK                                00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000030
* LOC,LTYP,ITYP,IBLKOT,MODE,IERR,IDES 00000040
COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH 00000050
COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW 00000060
COMMON/LABELS/STATRA(2,200),NLABEL 00000070
COMMON/FLOW/IFL,IRP 00000080
DIMENSION IPATH(100),ISTCK(100) 00000090
INTEGER FLWLST(100),BRANCH,STATRA 00000100
INTEGER BITPUT,BITGET 00000110
EQUIVALENCE (IPATH(1),A(1)),(ISTCK(1),A(101)),(FLWLST(1),A(201)) 00000120
IF (IFL .EQ. -1) GO TO 3000 00000130
IRX=0 00000140
NSTCK=0 00000150
NFLOW=0 00000160
NOC=0 00000170
NPTHS=0 00000180
IHLKST=1 00000190
CALL CHKLST 00000200
WRITE(6,8) 00000210
8 FORMAT(1H1,3RH***** RESULTS OF FLOW ANALYSIS *****//) 00000220
5 DO 10 I=1,NID 00000230
10 IDTBL(4,I)=IDTBL(8,I) 00000240
12 IF (NFLOW .EQ. 0) GO TO 20 00000250
NOC=0 00000260
DO 15 I=1,NFLOW 00000270
IF (IABS(FLWLST(I)) .NE. IHLKST) GO TO 15 00000280
NOC=NOC+1 00000290
15 CONTINUE 00000300
IF (NOC .GT. IRP) GO TO 1500 00000310
20 NFLOW=NFLOW+1 00000320
IF (NFLOW .GT. 100) GO TO 4000 00000330
FLWLST(NFLOW)=IHLKST 00000340
IEND=BITGET(IBLOCK(IHLKST),24,12)-1 00000350
IF (IEND .EQ. -1) IEND=NBLOCK 00000360
NBR=BITGET(IBLOCK(IHLKST),6,6) 00000370
ISTART=IEND-NBR+1 00000380
IBLKST=NXTBLK(ISTART,IEND) 00000390
IF (NBR .EQ. 1) GO TO 25 00000400
FLWLST(NFLOW)=-FLWLST(NFLOW) 00000410
NSTCK=NSTCK+1 00000420
IF (NSTCK .GT. 100) GO TO 5000 00000430
ISTCK(NSTCK)=NXTBLK(IEND,IEND) 00000440
IF (NBR .EQ. 2) GO TO 25 00000450
DO 22 J=3,NBR 00000460
NSTCK=NSTCK+1 00000470
IF (NSTCK .GT. 100) GO TO 5000 00000480
22 ISTCK(NSTCK)=NXTBLK(IEND-J+2,IEND) 00000490
25 IF (IHLKST .NE. 0) GO TO 12 00000500
NPATH=0 00000510
NPTHS=NPTHS+1 00000520
DO 1000 I=1,NFLOW 00000530
BRANCH=IABS(FLWLST(I)) 00000540
ISTART=BRANCH+1 00000550

```



NXBLOK=BITGET(IBLOCK(BRANCH),24,12)	00000560
IF(NXBLOK .EQ. 0) NXHLOK=NBLOCK*1	00000570
ISL=BITGET(IBLOCK(BRANCH),32,8)	00000580
ILOOP=BITGET(IBLOCK(BRANCH),12,6)	00000590
NHR=BITGET(IBLOCK(BRANCH),6,6)	00000600
IEND=NXBLOK-NHR-1	00000610
IF(ISL .EQ. 0) GO TO 45	00000620
NPATH=NPATH+1	00000630
IPATH(NPATH)=STATRA(1,ISL)	00000640
STATRA(2,ISL)=BITPUT(STATRA(2,ISL),1,18)	00000650
45 IF(IBLOCK(ISTART) .LT. 1000) GO TO 1000	00000660
DO 500 J=ISTART,IEND	00000670
IT=IBLOCK(J)/1000	00000680
LOC=IBLOCK(J)-IT*1000	00000690
GO TO(50,60,70,80,90,100,200),IT	00000700
50 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120	00000710
IF(IDTBL(4,LOC) .EQ. 2) GO TO 55	00000720
IF(IDTBL(4,LOC) .EQ. 4) GO TO 180	00000730
52 IDTBL(4,LOC)=1	00000740
30 IF(BITGET(IDTBL(3,LOC),17,1) .EQ. 0) GO TO 500	00000750
KTYPE=BITGET(IDTBL(3,LOC),10,3)	00000760
NXQV=LOC	00000770
53 NXQV=IDTBL(10,NXQV)	00000780
IF(NXQV .EQ. LOC) GO TO 500	00000790
IF(BITGET(IDTBL(3,NXQV),10,3) .EQ. KTYPE) GO TO 54	00000800
IDTBL(4,NXQV)=0	00000810
GO TO 53	00000820
54 IDTBL(4,NXQV)=1	00000830
GO TO 53	00000840
55 IF(ILOOP .EQ. 0) GO TO 52	00000850
57 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110	00000860
IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 52	00000870
ILOOP=ISTACK(3,ILOOP)	00000880
GO TO 57	00000890
60 IF(IDTBL(4,LOC) .EQ. 0) GO TO 140	00000900
IF(IDTBL(4,LOC) .EQ. 3) GO TO 130	00000910
GO TO 500	00000920
70 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120	00000930
IF(IDTBL(4,LOC) .EQ. 2) GO TO 75	00000940
72 IDTBL(4,LOC)=2	00000950
GO TO 500	00000960
75 IF(ILOOP .EQ. 0) GO TO 72	00000970
77 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110	00000980
IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 72	00000990
ILOOP=ISTACK(3,ILOOP)	00010000
GO TO 77	00010010
80 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120	00010020
IF(IDTBL(4,LOC) .EQ. 2) GO TO 85	00010030
82 IDTBL(4,LOC)=3	00010040
GO TO 500	00010050
85 IF(ILOOP .EQ. 0) GO TO 82	00010060
87 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110	00010070
IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 82	00010080
ILOOP=ISTACK(3,ILOOP)	00010090
GO TO 87	00010100



90 IF (IDTBL(4,LOC) .NE. 3) GO TO 150	00001110
GO TO 500	00001120
100 IDTBL(4,LOC)=0	00001130
GO TO 500	00001140
180 JLOOP=BITGET(IDTBL(3,LOC),32,9)	00001150
185 IF (ILOOP .EQ. JLOOP) GO TO 160	00001160
ILOOP=ISTACK(3,ILOOP)	00001170
IF (ILOOP .EQ. 0) GO TO 30	00001180
GO TO 185	00001190
200 IF (IDTBL(4,LOC) .EQ. 0) GO TO 140	00001200
IF (IDTBL(4,LOC) .EQ. 3) GO TO 130	00001210
IDTBL(4,LOC)=4	00001220
GO TO 500	00001230
110 IERC=67	00001240
GO TO 400	00001250
120 IERC=68	00001260
GO TO 400	00001270
130 IERC=69	00001280
GO TO 400	00001290
140 IERC=70	00001300
GO TO 400	00001310
150 IERC=71	00001320
GO TO 400	00001330
160 IERC=72	00001340
400 IRX=1	00001350
IF (BITGET(IDTBL(3,LOC),20,1) .EQ. 1) GO TO 500	00001360
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,20)	00001370
CALL ERROR(IERC,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)	00001380
IF (IERC .NE. 70 .OR. NPATH .EQ. 0) GO TO 500	00001390
WRITE(6,410) (IPATH(K),K=1,NPATH)	00001400
410 FORMAT(6X,15H ALONG THE PATH,(10I6))	00001410
500 CONTINUE	00001420
1000 CONTINUE	00001430
1500 IF (FLWLST(NFLOW) .GT. 0) GO TO 1600	00001440
IBLKST=IABS(ISTCK(NSTCK))	00001450
IF (ISTCK(NSTCK) .LT. 0) FLWLST(NFLOW)=-FLWLST(NFLOW)	00001460
NSTCK=NSTCK-1	00001470
NOC=0	00001480
GO TO 5	00001490
1600 NFLOW=NFLOW-1	00001500
IF (NFLOW .GT. 0) GO TO 1500	00001510
IF (NLABEL .EQ. 0) GO TO 2010	00001520
DO 2000 J=1,NLABEL	00001530
IF (BITGET(STATRA(2,J),6,6) .EQ. 28) GO TO 2000	00001540
IF (BITGET(STATRA(2,J),18,3) .EQ. 1) GO TO 2000	00001550
WRITE(6,1800) STATRA(1,J)	00001560
IRX=1	00001570
1800 FORMAT(6X,57H THERE IS NO COMPLETE PATH THAT CONTAINS STATEMENT NU	00001580
IMBER,I6)	00001590
2000 CONTINUE	00001600
2010 IF (IRX .EQ. 0) WRITE(6,2020)	00001610
2020 FORMAT(//6X,16H NO ERRORS FOUND)	00001620
WRITE(6,2100) NPTH	00001630
2100 FORMAT(/////6X,25H NUMBER OF PATHS CHECKED-,I6)	00001640
RETURN	00001650
3000 WRITE(6,3001)	00001660
3001 FORMAT(//31X,57H FLOW ANALYSIS WAS NOT PERFORMED DUE TO ERRORS IN	00001670
*PROGRAM)	00001680
IFL=IRP+1	00001690
RETURN	00001700
4000 WRITE(6,4001)	00001710
4001 FORMAT(//29X,63H TABLE OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS	00001720
\$IS TERMINATED)	00001730
RETURN	00001740
5000 WRITE(6,5001)	00001750
5001 FORMAT(//29X,63H STACK OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS	00001760
\$IS TERMINATED)	00001770
RETURN	00001780
END	00001790



```

SUBROUTINE FNCSTR
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IHLKOT,MODE,IERR,IDES
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
COMMON/FUNC/IFNCRA(5,22),MARG5,IARG5(50),FNCLOC(5),NFUNC
COMMON/BASRLK/IBLOCK(2500),NHLOCK,NB,NERNCH
INTEGER FNCLOC,BITPUT,BITGET
IF(NFUNC.EQ.0) RETURN
DO 40 I=1,NFUNC
LOC=FNCLOC(I)
IF(BITGET(IDTBL(3,LOC),19,1).EQ.1) GO TO 50
NARG=IFNCRA(I,1)
IVAR=0
ITP=BITGET(IDTBL(3,LOC),10,3)
IF(BITGET(IDTBL(3,LOC),18,1).EQ.1) GO TO 20
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)
DO 5 J=1,NLIST
IF(IDTBL(1,LOC).NE.ISUHLT(1,J)) GO TO 5
IF(IDTBL(2,LOC).NE.ISUHLT(2,J)) GO TO 5
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),J,32)
LISTLC=J
GO TO 21
5 CONTINUE
CALL ERROR(52,IDM1,IDM2,IDM3,IDM4)
NLIST=NLIST+1
IF(NLIST.GT.200) GO TO 60
ISUBLT(1,NLIST)=IDTBL(1,LOC)
ISUBLT(2,NLIST)=IDTBL(2,LOC)
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,32)
IPTR=NINTFC+1
IF(ITYP.EQ.8.AND.1.EQ.1) ITP=0
ISUBLT(3,NLIST)=BITPUT(0,ITP,13)
ISUBLT(3,NLIST)=BITPUT(ISUBLT(3,NLIST),NARG,6)
ISUBLT(4,NLIST)=IPTR
NINTFC=IPTR*(NARG-1)/3
IF(NINTFC.GT.600) GO TO 70
DO 10 J=IPTR,NINTFC
10 INTFAC(J)=IFNCRA(I,J-IPTR+2)
GO TO 40
20 LISTLC=BITGET(IDTBL(3,LOC),32,9)
21 ITP=ISUBLT(4,LISTLC)
IF(BITGET(ISUHLT(3,LISTLC),14,1).EQ.1) GO TO 22
NAR2=BITGET(ISUHLT(3,LISTLC),6,6)
IF(NARG.NE.NAR2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),17,18)
NARG5=MIN0(NARG,NAR2)
GO TO 24
22 IVAR=1
IF(NARG.LT.2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),17,18)
NARG5=NARG
ITP1=BITGET(INTFAC(IPTR),3,3)
NDIM1=BITGET(INTFAC(IPTR),6,3)
24 IF(ITYP.EQ.8.AND.1.EQ.1) GO TO 25
IF(BITGET(ISUHLT(3,LISTLC),10,4).EQ.4) GO TO 25
JTP=BITGET(ISUHLT(3,LISTLC),13,3)

```



IF(JTP .NE. ITP) CALL ERROR(49, IDTBL(1, LOC), IDTBL(2, LOC), I7, I8)	00000540
25 NDPTH=IPTR*(NARGS-1)/3	00000550
KOUNT=0	00000560
DO 32 K=IPTR, NDPTH	00000570
ICOL1=-6	00000580
ICOL2=-3	00000590
DO 32 J=1,3	00000600
KOUNT=KOUNT+1	00000610
IF(KOUNT .GT. NARGS) GO TO 40	00000620
ICOL1=ICOL1+9	00000630
ICOL2=ICOL2+9	00000640
IF(IVAR .EQ. 1) GO TO 26	00000650
ITP1=BITGET(INTFAC(K), ICOL1, 3)	00000660
NDIM1=BITGET(INTFAC(K), ICOL2, 3)	00000670
26 ITP2=BITGET(IFNCRA(I, K-IPTR+2), ICOL1, 3)	00000680
NDIM2=BITGET(IFNCRA(I, K-IPTR+2), ICOL2, 3)	00000690
IF(NDIM1 .NE. NDIM2) CALL ERROR(50, KOUNT, I7, I8, I9)	00000700
IF(ITP2 .EQ. 0) GO TO 32	00000710
IF(ITP1 .EQ. 0) GO TO 28	00000720
IF(ITP1 .NE. ITP2) CALL ERROR(51, KOUNT, I7, I8, I9)	00000730
GO TO 32	00000740
28 INTFAC(K)=BITPUT(INTFAC(K), ITP2, ICOL1)	00000750
32 CONTINUE	00000760
GO TO 40	00000770
50 CALL STFNC(I)	00000780
40 CONTINUE	00000790
RETURN	00000800
60 CALL ERROR(92, IDM1, IDM2, IDM3, IDM4)	00000802
STOP	00000804
70 CALL ERROR(93, IDM1, IDM2, IDM3, IDM4)	00000806
STOP	00000808
END	00000810



	SUBROUTINE FORM	00000010
	COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTP,LVVAL,	00000020
	•LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
	COMMON/LVTABL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSQSP( 1)	00000040
	COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING	00000050
	COMMON/VAR/VFOR,NCHAR,NCHARP,CHAR,NDICT	00000060
	COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG	00000070
	COMMON /STRING/ NTYPE,NSTR,STR	00000080
	INTEGER BITPUT,BITGET	00000090
	INTEGER VFOR(30),CHAR,STR(1)	00000100
	LOGICAL ERRFLG	00000110
	DATA IX/1HX/	00000120
C	EXECUTE	00000130
	GO TO 25000	00000140
25001	CONTINUE	00000150
	IF(CHAR .NE. IX) NDICT=-NDICT	00000160
	NCHARP=NCHARP+1	00000170
	STR(NCHARP)=NDICT	00000180
	IF(.NOT. ERRFLG) RETURN	00000190
	NCHAR=NCHAR+1	00000200
	NC=1+(NCHARP-1)/4	00000210
	ICHAR=BITGET(CHAR,8,8)	00000220
	VFOR(NC)=BITPUT(VFOR(NC),ICHAR,8*NCHAR)	00000230
	IF(NCHAR .EQ. 4) NCHAR=0	00000240
C	COMPLETE	00000250
	RETURN	00000260
25000	CONTINUE	00000270
	GO TO 25001	00000280
	END	00000290



SUBROUTINE FORMEL	00000010
COMMON A(1326),D(500),IDTHL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGIO,IDTYP,NIO,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
INTEGER B(50),L	00000050
DIMENSION IC(10)	00000060
DATA IC/75,77,78,91,92,93,96,97,107,126/	00000070
DATA IH/1HH/	00000080
GO TO (100,10,20,100,40,100,100,100),JTYP	00000090
10 CALL CAA(D,M,NXTID)	00000100
RETURN	00000110
20 DO 25 I=1,10	00000120
IF(D(I) .NE. IH) GO TO 25	00000130
CALL CAI(D,I-1,N2)	00000140
IF(N2 .LT. 1) GO TO 110	00000150
M=N2+I	00000160
IF(M .GT. 500) CALL ERROR(5,IDM1,IDM2,IDM3,IDM4)	00000170
JPTR=LSTART+M	00000180
IST=I+1	00000190
DO 22 J=IST,M	00000200
ICHAR=BITGET(D(J),8,8)	00000210
DO 23 II=1,10	00000220
IF(ICHAR .EQ. IC(II)) GO TO 120	00000230
23 CONTINUE	00000240
22 CONTINUE	00000250
IF(ITYP .EQ. 28) RETURN	00000260
IF(N2 .GT. 4) CALL ERROR(5,IDM1,IDM2,IDM3,IDM4)	00000270
RETURN	00000280
25 CONTINUE	00000290
CALL ERROR(3,IDM1,IDM2,IDM3,IDM4)	00000300
RETURN	00000310
40 CALL CAI(D,M,N2)	00000320
100 RETURN	00000330
110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000340
RETURN	00000350
120 CALL ERROR(23,IDM1,IDM2,IDM3,IDM4)	00000360
RETURN	00000370
END	00000380



SUBROUTINE FRMAT	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,	00000050
1 DIR,ICOM,ISEP	00000060
DIMENSION RPLOC(20),IALPH(6)	00000070
INTEGER A,RPLOC,AICH,RPAR,BLANK,SEPST,SEPND,DIR	00000080
DATA BLANK/1H /,AICH/1HH/,LPAR/1H(/,RPAR/1H)/	00000090
DATA IALPH/1HF,1HO,1HR,1HM,1HA,1HT/	00000100
IFRMT=0	00000110
DO 4 I=1,6	00000120
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 70	00000130
4 CONTINUE	00000140
NSTART=JPTR	00000150
IF(NEXT(JPTR) .NE. LPAR) GO TO 70	00000160
DO 10 I=1,N	00000170
IF(ITYP(JPTR) .EQ. 2) GO TO 1	00000180
IF(JPTR .GT. N) GO TO 12	00000190
GO TO 10	00000200
1 JPTR=JPTR-1	00000210
CALL GNLE	00000220
IF(JTYP .NE. 3) GO TO 10	00000230
J1=JPTR-1	00000240
IH=0	00000250
DO 5 J=LSTART,J1	00000260
IF(IH .EQ. 1) GO TO 3	00000270
IF(A(J) .EQ. AICH) IH=1	00000280
GO TO 5	00000290
3 A(J)=BLANK	00000300
5 CONTINUE	00000310
10 CONTINUE	00000320
12 NPAR=0	00000330
NRP=0	00000340
DO 20 I=NSTART,N	00000350
IF(A(I) .NE. LPAR) GO TO 15	00000360
NPAR=NPAR+1	00000370
IF(NPAR .GT. 3) GO TO 70	00000380
GO TO 20	00000390
15 IF(A(I) .NE. RPAR) GO TO 20	00000400
NPAR=NPAR-1	00000410
NRP=NRP+1	00000420
RPLOC(NRP)=I	00000430
IF(NPAR .LT. 0) GO TO 70	00000440
20 CONTINUE	00000450
IF(NPAR .NE. 0) GO TO 70	00000460
JPTR=RPLOC(NRP)+1	00000470
IF(NEXT(JPTR) .NE. BLANK) GO TO 70	00000480
DO 60 I=1,NRP	00000490
IGPND=RPLOC(I)	00000500
DO 25 J=1,N	00000510
K=IGPND-J	00000520
IF(A(K) .NE. LPAR) GO TO 25	00000530
IGPST=K	00000540
GO TO 30	00000550



25 CONTINUE	00000560
30 CALL GROUP	00000570
IF(IGRP .EQ. 0) RETURN	00000580
IF(I .EQ. NRP) GO TO 65	00000590
JPTR=IGPST-1	00000600
31 IPV=JPTR	00000610
IF(IPREV(IPV) .EQ. 2) GO TO 31	00000620
IGPST=JPTR+2	00000630
35 SEPST=IGPND+1	00000640
DIR=1	00000650
CALL SEPAR	00000660
IF(ISEP .NE. 1) GO TO 40	00000670
IGPND=SEPND	00000680
GO TO 50	00000690
40 JPTR=SEPST	00000700
IF(NEXT(JPTR) .NE. RPAR) GO TO 70	00000710
SEPST=IGPST-1	00000720
DIR=-1	00000730
CALL SEPAR	00000740
IF(ISEP .NE. 1) GO TO 45	00000750
IGPST=SEPND	00000760
GO TO 50	00000770
45 IF(A(SEPND) .NE. LPAR) GO TO 70	00000780
50 DO 55 J=IGPST,IGPND	00000790
A(J)=BLANK	00000800
55 CONTINUE	00000810
60 CONTINUE	00000820
65 IFRMT=1	00000830
RETURN	00000840
70 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000850
RETURN	00000860
END	00000870



```

SUBROUTINE GENFOL
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXITB(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IOTYP,NID,
* LOC,LTYP,ITYP,IBLKUT,MODE,IERR,IDES
COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)
COMMON/INPUT/NCALL,IN,IOP
COMMON/WASTE/IDUM(63)
DIMENSION JSES(2)
INTEGER BLKTBL,EXTTBL,HITGET
DATA JSES(1),JSES(2)/4HSESC,2HOM/
DATA ID/2HU,/,1J/2HU/
WRITE(6,2)
2 FORMAT(1H1)
IF(NBLK.EQ. 0) GO TO 6
K=-1
DO 5 I=1,NBLK
K=K+1
INDEX=BLKTBL(I)
ISZ=HITGET(ISUBLT(3,INDEX),32,15)
WRITE(IOP,3) ISUBLT(1,INDEX),ISUBLT(2,INDEX),K,ISZ
3 FORMAT(5X,4H COMMON/,A4,A2,3H/IX,I2,1H(,1b,1H))
5 CONTINUE
6 WRITE(IOP,7) MODE
7 FORMAT(5X,4H J=1/5X,6H MODE=,11/5X,13H DO 10 I=1,13/5X,6H J=J-1)
IF(NBLK.EQ. 0) GO TO 22
K=-1
DO 20 I=1,NBLK
K=K+1
KK=1000+K
INDEX=BLKTBL(I)
ISZ=HITGET(ISUBLT(3,INDEX),32,15)
WRITE(IOP,10) KK,ISZ,K,KK
10 FORMAT(5X,4H DO ,I4,5H K=1,,16/5X,3H IX,I2,5H(K)=1/IX,I4,
* 9H CONTINUE)
IF(ICOMP(JSES,ISUBLT,INDEX,4).EQ. 0) GO TO 20
WRITE(IOP,15) K,K,K
15 FORMAT(5X,3H IX,I2,7H(17)=10/5X,3H IX,I2,7H(20)=11/
* 5X,3H IX,I2,7H(23)=12)
20 CONTINUE
22 NARG=HITGET(IDTBL(3,1),7,6)
DO 30 I=1,NARG
IF(I.EQ. NARG) GO TO 25
IDUM(I)=IU
GO TO 30
25 IDUM(I)=IJ
30 CONTINUE
WRITE(IOP,35) IDTBL(1,1),IDTBL(2,1),(IDUM(I),I=1,NARG)
35 FORMAT(5X,6H CALL ,A4,A2,1H(,41A2/5X,1H1,1X,22A2)
WRITE(IOP,40)
40 FORMAT(5X,24H IF(MODE.EQ. 3) GO TO 5/5X,14H CALL MODID(J)/
* 3X,12H 5 ENDFILE 3/2X,12H 10 CONTINUE/5X,12H CALL CMPARE/
* 5X,10H REWIND 13/5X,10H REWIND 14/5X,10H REWIND 15/5X,5H STOP/
* 5X,4H END)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560

```



```

SUBROUTINE GLOTAB                                00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES 00000040
COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTHL(200),EXTTBL(100),ISUHS(100) 00000050
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600) 00000060
INTEGER BLKTBL,BITGET,EXTTBL,KLAS(5,7) 00000070
DATA KLAS/4HUSER,4H SUP,4HPLIE,1H0,1H ,4HSUBR,4HOUTI,4HNE M, 00000080
$ 4HODUL,1HE,4HFUNC,4HTION,4H MOD,3HULE,1H ,4HNCI,4HLLAR,4HY SU, 00000090
$ 4HBPR0,4HGRAM,4HANSI,4H FUN,4HCTIO,1HN,1H ,4HMAIN,4H PRO,4HGRAM, 00000100
$ 1H ,1H ,4HEXTR,4HAORD,4HINAR,4HY SU,4HBPR,/ 00000110
DATA IBLNK/1H / 00000120
WRITE(6,1) 00000130
1 FORMAT(1H1,4HX,23H GLOBAL REFERENCE TABLE) 00000140
IF(NREF.EQ. 0) GO TO 25 00000150
WRITE(6,2) 00000160
2 FORMAT(//50X,20H EXTERNAL REFERENCES) 00000170
DO 20 I=1,NREF 00000180
INDEX=EXTTBL(I) 00000190
J=BITGET(ISUHLT(3,INDEX),10,4) 00000200
WRITE(6,10) ISUHLT(1,INDEX),ISUHLT(2,INDEX),(KLAS(II,J+1),II=1,5) 00000210
10 FORMAT(45X,A4,A2,4X,5A4) 00000220
20 CONTINUE 00000230
25 IF(NBLK.EQ. 0 .OR. (NBLK.EQ. 1 .AND. ISUBLT(1,BLKTHL(1)) .EQ. 00000240
$ IBLNK)) GO TO 40 00000250
WRITE(6,30) 00000260
30 FORMAT(//49X,23H LABELLED COMMON BLOCKS/43X,11H BLOCK NAME,7X, 00000270
$ 5H SIZE,7X,6H CLASS) 00000280
DO 38 J=1,NBLK 00000290
INDEX=BLKTBL(J) 00000300
ICAT=BITGET(ISUHLT(3,INDEX),10,4)-6 00000310
ISZ=BITGET(ISUHLT(3,INDEX),32,15) 00000320
35 FORMAT(46X,A4,A2,5X,I8,5X,9HCATEGORY ,I2) 00000330
WRITE(6,35) ISUHLT(1,INDEX),ISUHLT(2,INDEX),ISZ,ICAT 00000340
38 CONTINUE 00000350
40 WRITE(6,45) 00000360
45 FORMAT(///49X,24H SUBROUTINES ENCOUNTERED) 00000370
DO 60 I=1,NSUBS 00000380
INDEX=ISUHS(I) 00000390
J=BITGET(ISUHLT(3,INDEX),10,4) 00000400
WRITE(6,10) ISUHLT(1,INDEX),ISUHLT(2,INDEX),(KLAS(II,J+1),II=1,5) 00000410
60 CONTINUE 00000420
RETURN 00000430
END 00000440

```



SUBROUTINE GNLE	00000010
COMMON A(1326),O(500),IDTBL(11,500),INITID(3),LASTID(3),ISFCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/LOGIC/LOG,LOGST	00000050
COMMON/REALND/IREAL,IRELND,IP	00000060
INTEGER A,D,H,HLANK,PLUS,EQUALS,SLASH,PPAR,COMMA,ASTRIK,	00000070
1 AICH,DECP	00000080
DATA BLANK/1H /,PLUS/1H+/,MINUS/1H-/,EQUALS/1H=/,SLASH/1H//,	00000090
• PPAR/1H)/,COMMA/1H/,ASTRIK/1H*/.AICH/1HH/,LPAK/1H(/,DECP/1H./	00000100
JTYP=0	00000110
NXT=NEXT(JPTR)	00000120
IF(NXT.EQ. HLANK) RETURN	00000130
LSTART=JPTR-1	00000140
IF(NXT.EQ. PLUS) GO TO 1	00000150
IF(NXT.EQ. PPAR) GO TO 1	00000160
IF(NXT.EQ. MINUS) GO TO 1	00000170
IF(NXT.EQ. SLASH) GO TO 1	00000180
IF(NXT.EQ. COMMA) GO TO 1	00000190
IF(NXT.EQ. EQUALS) GO TO 1	00000200
GO TO 2	00000210
1 JTYP=1	00000220
M=1	00000230
GO TO 90	00000240
2 IF(NXT.NE. ASTRIK) GO TO 4	00000250
IF(NEXT(JPTR).NE. ASTRIK) GO TO 1	00000260
M=2	00000270
JTYP=1	00000280
GO TO 90	00000290
4 IF(NXT.NE. LPAR) GO TO 40	00000300
IF(LSTART.EQ. 1) GO TO 10	00000310
IM1=LSTART-1	00000320
IF(IPREV(IM1).NE. 3) GO TO 1	00000330
10 JPTR=LSTART+1	00000340
NXT=NEXT(JPTR)	00000350
IF(NXT.EQ. HLANK) GO TO 120	00000360
IF(NXT.NE. PLUS.AND. NXT.NE. MINUS) GO TO 22	00000370
IP=JPTR	00000380
GO TO 24	00000390
22 IP=JPTR-1	00000400
24 CALL REALCK	00000410
IF(IREAL.EQ. 0) GO TO 1	00000420
IF(IDES.EQ. 1) GO TO 1	00000430
JPTR=IRELND+1	00000440
IF(NEXT(JPTR).NE. COMMA) GO TO 1	00000450
NXT=NEXT(JPTR)	00000460
IF(NXT.NE. PLUS.AND. NXT.NE. MINUS) GO TO 30	00000470
IP=JPTR	00000480
GO TO 35	00000490
30 IP=JPTR-1	00000500
35 CALL REALCK	00000510
IF(IREAL.EQ. 0) GO TO 120	00000520
IF(IDES.EQ. 1) GO TO 120	00000530
JPTR=IRELND+1	00000540
IF(NEXT(JPTR).NE. PPAR) GO TO 120	00000550



JTYP=6	00000560
M=JPTR-LSTART	00000570
GO TO 90	00000580
40 IF(NXT .NE. DECPY) GO TO 50	00000590
ITP=ITYPE(JPTR)	00000600
GO TO (42,44,120),ITP	00000610
42 LOGST=LSTART+1	00000620
CALL LOGCHK	00000630
IF(LOG .EQ. 0) GO TO 120	00000640
JTYP=7	00000650
M=JPTR-LSTART	00000660
GO TO 90	00000670
44 IP=LSTART	00000680
CALL REALCK	00000690
IF(IREAL .EQ. 0) GO TO 120	00000700
JTYP=4	00000710
M=IRELND-LSTART+1	00000720
GO TO 90	00000730
50 JPTR=LSTART	00000740
IF(ITYPE(JPTR) .NE. 2) GO TO 85	00000750
IF(ITYP .EQ. 28) GO TO 54	00000760
IP=LSTART	00000770
CALL REALCK	00000780
IF(IREAL .EQ. 0) GO TO 54	00000790
JTYP=4	00000800
M=IRELND-LSTART+1	00000810
GO TO 90	00000820
54 JPTR=LSTART+1	00000830
55 IF(ITYPE(JPTR) .EQ. 2) GO TO 57	00000840
GO TO 65	00000850
57 IF(JPTR .GT. N) GO TO 60	00000860
GO TO 55	00000870
60 M=N-LSTART+1	00000880
JTYP=5	00000890
GO TO 90	00000900
65 IF(A(JPTR-1) .NE. AICH) GO TO 67	00000910
IF(ITYP .EQ. 8 .OR. IITYP .EQ. 28 .OR. IITYP .EQ. 27) GO TO 70	00000920
67 M=JPTR-LSTART-1	00000930
JTYP=5	00000940
GO TO 90	00000950
70 IF(JPTR .GT. N) GO TO 120	00000960
M=N-LSTART+1	00000970
IF(M .GT. 500) M=500	00000980
JTYP=3	00000990
GO TO 90	00010000
85 JPTR=LSTART	00010010
IF(ITYPE(JPTR) .NE. 1) GO TO 120	00010020
IF(ITYP .EQ. 28) GO TO 100	00010030
88 CONTINUE	00010040
IF(ITYPE(JPTR) .NE. 3) GO TO 86	00010050
M=JPTR-LSTART-1	00010060
JTYP=2	00010070
GO TO 90	00010080
86 IF(JPTR .GT. N) GO TO 87	00010090
GO TO 88	00010100



100 M=1	00001110
JTYPE=2	00001120
GO TO 90	00001130
87 M=N-LSTART+1	00001140
JTYPE=2	00001150
90 CONTINUE	00001160
DO 91 L=1,M	00001170
LL=LSTART+L-1	00001180
D(L)=A(LL)	00001190
91 CONTINUE	00001200
JPTR=LSTART+M	00001210
CALL SQUEEZ	00001220
CALL FORMEL	00001230
RETURN	00001240
120 CONTINUE	00001250
JTYPE=8	00001260
RETURN	00001270
END	00001280

SUBROUTINE GOTO	00000010
COMMON A(I326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISHCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYPE,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYPE,ITYP,IBLKUT,MODE,IEER,IDES	00000040
COMMON/LABELS/STATRA(2,200),NLABEL	00000050
COMMON/DASHLK/IBLOCK(2500),NBLOCK,NB,NRRNCH	00000060
DIMENSION IALPH(4)	00000070
INTEGER STATRA,BLANK	00000080
INTEGER BITPUT	00000090
DATA IALPH/1HG,1HO,1HT,1HO/	00000100
DATA BLANK/1H /	00000110
DO 5 I=1,4	00000120
IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 10	00000130
5 CONTINUE	00000140
CALL GNLE	00000150
IF(JTYPE .NE. 5) GO TO 10	00000160
CALL STSCH	00000170
STATRA(2,LOC)=BITPUT(STATRA(2,LOC)+1,12)	00000180
IF(NEXT(JPTR) .NE. BLANK) GO TO 10	00000190
NBLOCK=NBLOCK+1	00000200
IBLOCK(NBLOCK)=LOC	00000210
NRRNCH=1	00000220
NH=1	00000230
RETURN	00000240
10 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000250
RETURN	00000260
END	00000270



SUBROUTINE GROUP	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,UTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,	00000050
1 DIR,ICOM,ISEP	00000060
INTEGER A,RPAR,SEPST,SEPND,DIR	00000070
DATA RPAR/(1H)/	00000080
JPTR=IGPST+1	00000090
IF(NEXT(JPTR) .EQ. RPAR) GO TO 20	00000100
SEPST=JPTR-1	00000110
DIR=1	00000120
CALL SEPAR	00000130
IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30	00000140
IF(ISEP .EQ. 0) IDESST=SEPST	00000150
IF(ISEP .EQ. 1) IDESST=SEPND+1	00000160
JPTR=IDESST	00000170
IF(NEXT(JPTR) .EQ. RPAR) GO TO 20	00000180
SEPST=IGPND-1	00000190
DIR=-1	00000200
CALL SEPAR	00000210
IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30	00000220
DIR=1	00000230
10 CONTINUE	00000240
CALL UDESCR	00000250
IF(IDES .EQ. 0) GO TO 40	00000260
SEPST=IDESND+1	00000270
JPTR=SEPST	00000280
IF(NEXT(JPTR) .EQ. RPAR) GO TO 20	00000290
CALL SEPAR	00000300
IF(ISEP .EQ. 0 .OR. ISEP .EQ. -1) GO TO 30	00000310
IDESST=SEPND+1	00000320
JPTR=IDESST	00000330
IF(NEXT(JPTR) .NE. RPAR) GO TO 10	00000340
20 IGRP=1	00000350
RETURN	00000360
30 IGRP=0	00000370
CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000380
RETURN	00000390
40 CALL ERROR(8,IDESST,IDM2,IDM3,IDM4)	00000400
IGRP=0	00000410
RETURN	00000420
END	00000430



SUBROUTINE GRT	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)	00000050
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)	00000060
INTEGER EXTBL,BLKTBL,BLANK,BITPUT,BITGET	00000070
DATA BLANK/1H /	00000080
WRITE(6,1)	00000090
1 FORMAT(//)	00000100
ISUB=INITID(2)	00000110
IF (ISUB .EQ. 0) GO TO 15	00000120
10 ISUB=IDTBL(4,ISUB)	00000130
IF (ISUB .EQ. 0) GO TO 15	00000140
IF (BITGET(IDTBL(3,ISUB),19,1) .EQ. 1) GO TO 10	00000150
IF (NREF .EQ. 0) GO TO 4	00000160
DO 3 K=1,NREF	00000170
INDEX=EXTTBL(K)	00000180
IF (IDTBL(1,ISUB) .EQ. ISUBLT(1,INDEX) .AND. IDTBL(2,ISUB) .EQ.	00000190
ISUBLT(2,INDEX)) GO TO 10	00000200
3 CONTINUE	00000210
4 NREF=NREF+1	00000220
IF (NREF .GT. 100) GO TO 50	00000230
EXTTBL(NREF)=BITGET(IDTBL(3,ISUB),32,9)	00000240
IF (MODE .EQ. 1) GO TO 10	00000250
KLAS=BITGET(ISUBLT(3,EXTTBL(NREF)),10,4)	00000260
IF (KLAS.EQ.1.OR.KLAS.EQ.2) WRITE(9) IDTBL(1,ISUB),IDTBL(2,ISUB)	00000270
GO TO 10	00000280
15 IBLK=INITID(3)	00000290
20 IF (IBLK .EQ. 0) RETURN	00000300
IF (IDTBL(1,IBLK) .EQ. BLANK) GO TO 45	00000310
DO 25 I=1,NLIST	00000320
IF (IDTBL(1,IBLK) .NE. ISUBLT(1,I)) GO TO 25	00000330
IF (IDTBL(2,IBLK) .NE. ISUBLT(2,I)) GO TO 25	00000340
LISTLC=I	00000350
IF (BITGET(ISUBLT(3,I),10,4) .EQ. 7) GO TO 30	00000360
IF (BITGET(ISUBLT(3,I),32,15) .NE. 0) GO TO 30	00000370
ISZ=IDTBL(5,IBLK)	00000380
ISUBLT(3,I)=BITPUT(ISUBLT(3,I),ISZ,32)	00000390
GO TO 30	00000400
25 CONTINUE	00000410
CALL ERROR(62,IDTBL(1,IBLK),IDTBL(2,IBLK),IDM3,IDM4)	00000420
NLIST=NLIST+1	00000430
ISUBLT(1,NLIST)=IDTBL(1,IBLK)	00000440
ISUBLT(2,NLIST)=IDTBL(2,IBLK)	00000450
ISZ=IDTBL(5,IBLK)	00000460
ISUBLT(3,NLIST)=BITPUT(ISZ,10,10)	00000470
LISTLC=NLIST	00000480
30 IDTBL(3,IBLK)=BITPUT(IDTBL(3,IBLK),LISTLC,32)	00000490
IF (NBLK .EQ. 0) GO TO 40	00000500
DO 35 K=1,NBLK	00000510
IF (LISTLC .EQ. BLKTBL(K)) GO TO 45	00000520
35 CONTINUE	00000530
40 NBLK=NBLK+1	00000540
IF (NBLK .GT. 200) GO TO 60	00000550
BLKTBL(NBLK)=LISTLC	00000560
45 IBLK=IDTBL(4,IBLK)	00000570
GO TO 20	00000580
50 CALL ERROR(59,IDM1,IDM2,IDM3,IDM4)	00000590
STOP	00000600
60 CALL ERROR(60,IDM1,IDM2,IDM3,IDM4)	00000610
STOP	00000620
END	00000630



FUNCTION ICOMP(IVAR,JVAR,NVAR,NROW)	00000010
DIMENSION IVAR(1),JVAR(NROW,1)	00000020
ICOMP=0	00000030
IF(IVAR(1) .NE. JVAR(1,NVAR)) RETURN	00000040
IF(IVAR(2) .NE. JVAR(2,NVAR)) RETURN	00000050
ICOMP=1	00000060
RETURN	00000070
END	00000080

SUBROUTINE IMPTYP	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
DIMENSION IALPH(6)	00000050
INTEGER D,BITPUT,BITGET	00000060
DATA IALPH/1H1,1HJ,1HK,1HL,1HM,1HN/	00000070
IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 20	00000080
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)	00000090
DO 10 I=1,6	00000100
IF(D(I) .NE. IALPH(I)) GO TO 10	00000110
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),4,10)	00000120
GO TO 20	00000130
10 CONTINUE	00000140
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,10)	00000150
20 IF(ITYP .LE. 18) IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,21)	00000160
RETURN	00000170
END	00000180



```

SUBROUTINE INIT                                00000010
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES 00000040
COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC 00000050
COMMON/STRING/NTYPE,NSTR,STR(500) 00000060
COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP 00000070
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600) 00000080
COMMON/BASBLK/IBLOCK(2500),NELOCK,NB,NBRNCH 00000090
COMMON/STFUNC/NSTFNC,ISTFNC(10) 00000100
INTEGER RHSTYP 00000110
INTEGER A,EQUALS,COMMA,RPAR,STR 00000120
INTEGER HITPUT,HITGET,FNCLOC 00000130
DATA EQUALS/1H=/,LPAR/1H(/,COMMA/1H(/,RPAR/1H)/ 00000140
IFN=0 00000150
NTYPE=1 00000160
JPTR=JPTR 00000170
CALL GNLE 00000180
IF(JTYP,NE,2) GO TO 40 00000190
CALL SEARCH 00000200
IF(NEXT(JPTR),NE,EQUALS) GO TO 6 00000210
LOC2=0 00000220
IF(ISRCH(2),NE,1) GO TO 2 00000230
IF(LOC,NE,IFNCNM) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4) 00000240
IFN=1 00000250
CALL IMPTYP 00000260
LOC2=LOC 00000270
2 IF(ISRCH(1),NE,1) GO TO 18 00000280
IF(LOC2,EQ,0) GO TO 4 00000290
LOC=IDES 00000300
GO TO 5 00000310
18 IDTYP=1 00000320
CALL STORE 00000330
LOC=NID 00000340
IF(LOC2,EQ,0) GO TO 4 00000350
IDTBL(3,LOC)=IDTBL(3,LOC2) 00000360
GO TO 5 00000370
4 CALL IMPTYP 00000380
5 LHSTYP=BITGET(IDTBL(3,LOC),10,3) 00000390
IF(LHSTYP,EQ,5) NTYPE=2 00000400
GO TO 30 00000410
6 IF(A(JPTR-1),NE,LPAR) GO TO 40 00000420
IF(ISRCH(1),EQ,1) GO TO 12 00000430
IF(ISRCH(2),NE,1) GO TO 15 00000440
CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4) 00000450
GO TO 8 00000460
15 IDTYP=2 00000470
CALL STORE 00000480
LOC=NID 00000490
GO TO 8 00000500
7 CALL SWITCH 00000510
8 CALL IMPTYP 00000520
LHSTYP=BITGET(IDTBL(3,LOC),10,3) 00000530
IF(LHSTYP,EQ,5) NTYPE=2 00000540
NARG=0 00000550

```



ITYP=35	00000560
NSTFNC=NSTFNC+1	00000570
IF(NSTFNC.GT. 10) GO TO 60	00000575
ISTFNC(NSTFNC)=LOC	00000580
LOC1=LOC	00000590
IDTBL(3,LOC)=HITPUT(IDTBL(3,LOC),1,19)	00000600
10 CALL GNLE	00000610
IF(JTYP.NE. 2) GO TO 50	00000620
NARG=NARG+1	00000630
CALL SEARCH	00000640
IF(ISRCH(2).EQ. 1) CALL ERROR(54,NARG,1DM2,1DM3,1DM4)	00000650
IF(ISRCH(1).EQ. 1) GO TO 20	00000660
IDTYP=1	00000670
CALL STORE	00000680
LOC=NIU	00000690
GO TO 25	00000700
20 IF(BITGET(IDTBL(3,LOC),1,1).EQ. 1) CALL ERROR(54,NARG,17,18,19)	00000710
25 CALL IMPTYP	00000720
IF(NEXT(JPTR).EQ. COMMA) GO TO 10	00000730
IF(A(JPTH-1).NE. RPAR) GO TO 40	00000740
IDTBL(3,LOC1)=BITPUT(IDTBL(3,LOC1),NARG,7)	00000750
IF(NEXT(JPTR).EQ. EQUALS) GO TO 32	00000760
GO TO 40	00000770
12 IF(BITGET(IDTBL(3,LOC),1,1).NE. 1) GO TO 7	00000780
CALL IMPTYP	00000790
LHSTYP=BITGET(IDTBL(3,LOC),10,3)	00000800
IF(LHSTYP.EQ. 5) NTYPE=2	00000810
JPTR=IPTR	00000820
JBLOCK=NBLOCK+1	00000830
CALL EXPR	00000840
CALL PARSE	00000850
CALL HLKSTR	00000860
IBLOCK(JBLOCK)=IBLOCK(JBLOCK) - 1000	00000870
GO TO 32	00000880
30 NBLOCK=NBLOCK+1	00000890
JBLOCK=NBLOCK	00000900
IBLOCK(NBLOCK)=1000+LOC	00000910
32 NTMS=0	00000920
IPTR=JPTR	00000930
CALL EXPR	00000940
IF(JPTR.LE.N) CALL ERROR(7,1DM1,1DM2,1DM3,1DM4)	00000950
CALL PARSE	00000960
CALL FNCSTR	00000970
CALL EXPRCK	00000980
IF(ITYP.EQ. 35) GO TO 36	00000990
CALL HLKSTR	00010000
IBLOCK(NBLOCK+1)=IBLOCK(JBLOCK)	00010010
DO 34 K=JBLOCK,NBLOCK	00010020
34 IBLOCK(K)=IBLOCK(K+1)	00010030
IF(LTYP.EQ. 1) RETURN	00010040
36 IF(MODE.NE. 1) GO TO 35	00010050
IF(RHSTYP.EQ. 3.OR. PHSTYP.EQ. 4) RETURN	00010060
IF(INSTK.LT. 6) RETURN	00010070
JPTR=IPTR-1	00010080
CALL CNVRT	00010090



RETURN	00001100
35 IF(NFUNC .EQ. 0) RETURN	00001110
IF(IFN .EQ. 1) RETURN	00001120
DO 38 J=1,NFUNC	00001130
LOC=FNCLOC(J)	00001140
INDEX=BITGET(IDTBL(3,LOC),32,9)	00001150
KLAS=BITGET(ISUBLT(3,INDEX),10,4)	00001160
IF(KLAS .NE. 1 .AND. KLAS .NE. 2) GO TO 38	00001170
CALL CALL2	00001180
RETURN	00001190
38 CONTINUE	00001200
RETURN	00001210
40 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00001220
RETURN	00001230
50 CALL ERROR(15,IDM1,IDM2,IDM3,IDM4)	00001240
RETURN	00001250
60 CALL ERROR(89,IDM1,IDM2,IDM3,IDM4)	00001260
RETURN	00001270
END	00001280

SUBROUTINE INTRIN	00000010
COMMON A(1326),D(500),IDTRL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
SLOC,LTYP,ITYP,I8LKDT,MODE,IERR,IDES	00000040
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600)	00000050
INTEGER BITGET	00000060
DO 100 I=1,NLIST	00000070
IF(BITGET(ISUBLT(3,I),10,4) .NE. 4) GO TO 100	00000080
NXTID(1)=ISUBLT(1,I)	00000090
NXTID(2)=ISUBLT(2,I)	00000100
CALL SEARCH	00000110
CALL COMSCH	00000120
IF(ISRCH(1) .EQ. 1 .OR. ISRCH(3) .EQ. 1)	00000130
\$ CALL ERROR(74,NXTID(1),NXTID(2),IDM3,IDM4)	00000140
100 CONTINUE	00000150
RETURN	00000160
END	00000170



SUBROUTINE IO	00000010
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKOT,MODE,IERR,IDES	00000040
COMMON/STRING/NTYPE,NSTR,STR(500)	00000050
COMMON/LABELS/STATRA(2,200),NLABEL	00000060
COMMON/HASHLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000070
COMMON/INPOUT/NCALL,IN,IOP	00000080
DIMENSION IALPH1(4),IALPH2(5),IALPH3(8)	00000090
INTEGER A,STATRA,RPAR,COMMA,BLANK	00000100
INTEGER BITPUT,BITGET	00000110
DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H(/,BLANK/1H /,IC/1HC/	00000120
DATA IALPH1/1HR,1HE,1HA,1HD/	00000130
DATA IALPH2/1HW,1HR,1HI,1HT,1HE/	00000140
DATA IALPH3/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE/	00000150
IFPMT=0	00000160
IF(ITYP .EQ. 12) GO TO 10	00000170
DO 5 I=1,4	00000180
IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50	00000190
5 CONTINUE	00000200
GO TO 20	00000210
10 DO 15 I=1,5	00000220
IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50	00000230
15 CONTINUE	00000240
20 IF(NEXT(JPTR) .NE. LPAR) GO TO 50	00000250
CALL GNLE	00000260
IF(JTYP .EQ. 2) GO TO 22	00000270
CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)	00000280
GO TO 24	00000290
22 CALL SEARCH	00000300
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000310
IF(ISRCH(1) .EQ. 1) GO TO 25	00000320
IDTYP =1	00000330
CALL STORE	00000340
LOC=NID	00000350
25 CALL IMPTYP	00000360
IF(HITGET(IDTBL(3,LOC),1,1) .EQ. 1)	00000370
5 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00000380
IF(HITGET(IDTBL(3,LOC),10,3) .NE. 4)	00000390
5 CALL ERROR(22,IDM1,IDM2,IDM3,IDM4)	00000400
NBLOCK=NBLOCK+1	00000410
IBLOCK(NBLOCK)=2000+LOC	00000420
28 IF(NEXT(JPTR) .EQ. COMMA) GO TO 40	00000430
JPTR=JPTR+1	00000440
GO TO 30	00000450
40 CALL GNLE	00000460
IF(JTYP .NE. 5) GO TO 26	00000470
CALL STSRCH	00000480
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)	00000490
GO TO 29	00000500
26 IF(JTYP .NE. 2) GO TO 50	00000510
CALL SEARCH	00000520
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM1,IDM2)	00000530
IF(ISRCH(1) .EQ. 1) GO TO 27	00000540
IDTYP=1	00000550



CALL STORE	00000560
LOC=NI0	00000570
27 CALL IMPTYP	00000580
IF (HITGET(IDTHL(3,LOC),1,1) .NE. 1)	00000590
\$ CALL ERROR(43,IDM1,IDM2,IDM3,IDM4)	00000600
29 IFRMT=1	00000610
30 IF (NEXT(JPTR) .NE. RPAR) GO TO 50	00000620
IF (NEXT(JPTR) .NE. BLANK) GO TO 35	00000630
IF (ITYP .EQ. 12 .AND. IFRMT .EQ. 0)	00000640
\$ CALL ERROR(44,IDM1,IDM2,IDM3,IDM4)	00000650
GO TO 36	00000660
35 JPTR=JPTR-1	00000670
CALL EXPR	00000680
NTYPE=3	00000690
CALL PARSE	00000700
CALL IOSTR	00000710
36 IF (MODE .NE. 1 .AND. ITYP .EQ. 11) GO TO 37	00000720
RETURN	00000730
37 IF (ITYPE(1) .EQ. 2) GO TO 42	00000740
A(1)=IC	00000750
RETURN	00000760
42 WRITE(IOP,45) (A(I),I=1,6),(IALPH3(I),I=1,8)	00000770
45 FORMAT(72A1)	00000780
A(1)=IC	00000790
DO 47 I=2,6	00000800
47 A(I)=BLANK	00000810
RETURN	00000820
50 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000830
RETURN	00000840
END	00000850



```

SUBROUTINE IOSTR
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
$LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/FUNC/IFNCHA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC
COMMON/BASRLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
INTEGER BITGET
DO 100 I=1,MARGS
NBLOCK=NBLOCK+1
IVR=I
ICOL=9
LOC=BITGET(IARGS(IVR),ICOL,9)
ISUB=BITGET(IARGS(IVR),ICOL+2,2)
IF (ISUB .EQ. 1) GO TO 90
IF (ISUB .EQ. 2) GO TO 20
IF (ITYP .EQ. 11) GO TO 80
IF (ITYP .EQ. 12) GO TO 90
20 IDEF=BITGET(IARGS(IVR),ICOL+7,5)
NMOVE=I-IDEF-1
DO 30 J=1,NMOVE
ITEMP=NBLOCK-J
30 IBLOCK(ITEMP+1)=IBLOCK(ITEMP)
IBLOCK(ITEMP)=1000+LOC
NBLOCK=NBLOCK+1
IBLOCK(NBLOCK)=6000+LOC
GO TO 100
80 IBLOCK(NBLOCK)=1000+LOC
GO TO 100
90 IBLOCK(NBLOCK)=2000+LOC
100 CONTINUE
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320

```

```

FUNCTION IPREV(IA)
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
INTEGER A,BLANK
DATA BLANK/1H /
DO 10 I=1,N
J=IA-I+1
IF (J .EQ. 0) GO TO 20
IF (A(J) .EQ. BLANK) GO TO 10
IPREV=ITYP(J)
JPTR=IA-I
RETURN
10 CONTINUE
20 IPREV=3
JPTR=0
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180

```



```

FUNCTION ITYPE(ID)
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IEERR,IDES
INTEGER BITGET
NXT=NEXT(ID)
IVAL=BITGET(NXT,8,8)
IF(IVAL .GE. 193 .AND. IVAL .LE. 201) GO TO 10
IF(IVAL .GE. 209 .AND. IVAL .LE. 217) GO TO 10
IF(IVAL .GE. 226 .AND. IVAL .LE. 233) GO TO 10
IF(IVAL .GE. 240 .AND. IVAL .LE. 249) GO TO 20
ITYPE=3
RETURN
10 ITYPE=1
RETURN
20 ITYPE=2
RETURN
END

```

00000010  
00000020  
00000030  
00000040  
00000050  
00000060  
00000070  
00000080  
00000090  
00000100  
00000110  
00000120  
00000130  
00000140  
00000150  
00000160  
00000170  
00000180

```

SUBROUTINE LOGCHK
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IEERR,IDES
COMMON/LOGIC/LOG,LOGST
DIMENSION LOGOP(2,11),LOGRA(5),ILOG(2)
DATA LOGOP/2HLT,1H,2HLE,1H,2HGT,1H,2HGE,1H,2HEQ,1H,2HNE,1H,
* 2HOR,1H,3HAND,1H,3HNOT,1H,4HTRUE,1H,4HFALS,1H/
DATA IDEC/1H./
JPTR=LOGST
DO 10 I=1,6
NXT=NEXT(JPTR)
IF(NXT .EQ. IDEC) GO TO 12
10 LOGRA(I)=NXT
GO TO 20
12 IF(I .LT. 3) GO TO 20
CALL CAA(LOGRA,I-1,ILOG)
DO 15 I=1,11
IF(ICOMP(ILOG,LOGOP,I,2) .EQ. 1) GO TO 30
15 CONTINUE
20 LOG=0
RETURN
30 LOG=1
LOGID=I
RETURN
END

```

00000010  
00000020  
00000030  
00000040  
00000050  
00000060  
00000070  
00000080  
00000090  
00000100  
00000110  
00000120  
00000130  
00000140  
00000150  
00000160  
00000170  
00000180  
00000190  
00000200  
00000210  
00000220  
00000230  
00000240  
00000250  
00000260



SUBROUTINE LOGIF	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISPRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/STRING/NTYPE,NSTR,STR(500)	00000050
COMMON/BASHLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000060
INTEGER A,STR,AY,EF	00000070
DATA LPAR/1H(/,AY/1H(/,EF/1H(/	00000080
IF(NEXT(JPTR) .NE. AY) GO TO 110	00000090
IF(NEXT(JPTR) .NE. EF) GO TO 110	00000100
IF(NEXT(JPTR) .NE. LPAR) GO TO 110	00000110
JPTR=JPTR+1	00000120
CALL EXPR	00000130
NSTR=NSTR+1	00000140
STR(NSTR)= -5	00000150
NTYPE=2	00000160
CALL PARSE	00000170
CALL FNCSTR	00000180
CALL BLKSTR	00000190
IF(ITYP .GT. 15) GO TO 130	00000200
LTYP=1	00000210
GO TO (10,20,30,40,50,60,70,80,70,70,90,90,100,100,100),ITYP	00000220
10 CALL INIT	00000230
RETURN	00000240
20 CALL ASSIGN	00000250
RETURN	00000260
30 CALL GOTO	00000270
35 NBLOCK=NBLOCK+1	00000280
IBLOCK(NBLOCK)=998	00000290
NBRNCH=2	00000300
RETURN	00000310
40 CALL ASGOTO	00000320
45 NBLOCK=NBLOCK+1	00000330
IBLOCK(NBLOCK)=998	00000340
NBRNCH=NBRNCH+1	00000350
RETURN	00000360
50 CALL CTGOTO	00000370
GO TO 45	00000380
60 CALL APIF	00000390
GO TO 45	00000400
70 CALL SIMP	00000410
IF(ITYP .EQ. 7) RETURN	00000420
GO TO 35	00000430
80 CALL CALL	00000440
RETURN	00000450
90 CALL IO	00000460
RETURN	00000470
100 CALL AUXIO	00000480
RETURN	00000490
110 CALL ERROR(7,IDM1,IDM2,IDM3,IDM4)	00000500
RETURN	00000510
130 CALL ERROR(45,IDM1,IDM2,IDM3,IDM4)	00000520
RETURN	00000530
END	00000540



SUBROUTINE LOOPCK	00000010
COMMON/LABELS/STATRA(2,200),NLABEL	00000020
COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP	00000030
COMMON/HASHLK/IHLOCK(2500),NBLOCK,NB,NBRNCH	00000040
INTEGER STATRA,HITGET	00000050
IF(NSTACK .EQ. 0) RETURN	00000060
IHLKST=1	00000070
10 IHLKND=BITGET(IHLOCK(IHLKST),24,12)-1	00000080
LOOP2=BITGET(IHLOCK(IHLKST),12,6)	00000090
NBR=BITGET(IHLOCK(IHLKST),6,6)	00000100
IF(IHLKND .EQ. -1) IHLKND=NBLOCK	00000110
IST=IHLKND-NBR+1	00000120
DO 100 I=IST,IHLKND	00000130
JLOOP=LOOP2	00000140
IF(IHLOCK(I) .GE. 998) GO TO 100	00000150
IBLK=IHLOCK(I)	00000160
NXTBLK=BITGET(STATRA(2,IBLK),32,14)	00000170
KLOOP=BITGET(IHLOCK(NXTBLK),12,6)	00000180
IF(KLOOP .EQ. 0) GO TO 100	00000190
IF(JLOOP .EQ. 0) GO TO 200	00000200
50 IF(JLOOP .EQ. KLOOP) GO TO 100	00000210
JLOOP=ISTACK(3,JLOOP)	00000220
IF(JLOOP .EQ. 0) GO TO 200	00000230
GO TO 50	00000240
200 IBLK=IHLOCK(I)	00000250
WRITE(6,201) STATRA(1,IBLK)	00000260
201 FORMAT(6X,65H ILLEGAL TRANSFER INTO THE RANGE OF A DO LOOP AT STATEMENT NUMBER,I6)	00000270
100 CONTINUE	00000280
IF(IHLKND .EQ. NBLOCK) RETURN	00000290
IHLKST=IHLKND+1	00000300
GO TO 10	00000310
END	00000320
	00000330



	SUBROUTINE LVDLET	00000010
	COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL	00000020
	* IDSTRY,IVAL(10),ITYP1(10),NSKIP	00000030
	INTEGER FLGSPC,FL0MSK,FL1MSK,FL2MSK,FL5MSK,FL67,REGASP,THIS	00000040
	* ,FL3MSK,FL4MSK,SEQSPC	00000050
	COMMON/LVVTR1/MEMSZE,REGASP,NODSPC( 1)/LVVTR2/LSTSPC( 1)/	00000060
	*LVVTR3/LNKSPC( 1)/LVVTR4/FLGSPC( 1)	00000070
	COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FL67	00000080
	COMMON /LVTAHL/ MAPSZ,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)	00000090
	EQUIVALENCE (ILOG,RLOG)	00000100
	DATA NFLG02/45/	00000110
	KONFLC=0	00000120
C	DETERMINE DIRECTION TO PROCEED FOR MULTIVALUE LISTS	00000130
	IPOS=IPOS	00000140
	IPOS=IABS(IPOS)	00000150
	IF (IADD.NE.-1) GO TO 74	00000160
	IF (IARG.EQ.-1) GO TO 66	00000170
	IADD=IFUNC+IARG	00000180
	IF (IADD.GT.MEMSZE) IADD=IADD-MEMSZE	00000190
	IF (AND (FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 99	00000200
1	IF (NODSPC(IADD).EQ.IARG) GO TO 4	00000210
C	SEARCH CONFLICT LIST FOR THE FUNCTION	00000220
	IADD=LNKSPC(IADD)	00000230
	IF (AND (FLGSPC(IADD),FL5MSK) .NE. 0.) GO TO 99	00000240
	GO TO 1	00000250
66	IADD=IFUNC	00000260
C	TO DELETE A SPECIFIC TYPE OF NODE (INDEXED DELETE), GO TO 72	00000270
4	IF (ITYP.NE.-1) GO TO 72	00000280
	IF (AND (FLGSPC(IADD),FL0MSK) .EQ. 0.) GO TO 6	00000290
	ISADD=LSTSPC(IADD)	00000300
C	DELETE ENTIRE MULTIVALUED FUNCTION, RETRIEVE FIRST VALUE	00000310
	IVAL=NODSPC(ISADD)	00000320
5	NXTADD=LSTSPC(ISADD)	00000330
	NODSPC(ISADD)=NODSPC(REGASP)	00000340
	LSTSPC(ISADD)=REGASP	00000350
	LNKSPC(ISADD)=0	00000360
	FLGSPC(ISADD)=0	00000370
	ISUB=NODSPC(REGASP)	00000380
	LSTSPC(ISUB)=ISADD	00000390
	NODSPC(REGASP)=ISADD	00000400
	IF (AND (FLGSPC(NXTADD),FL0MSK) .NE. 0.) GO TO 2	00000410
	ISADD=NXTADD	00000420
	GO TO 5	00000430
C	FUNCTION IS SINGLE VALUED, RETRIEVE VALUE	00000440
6	IVAL=LSTSPC(IADD)	00000450
2	IF (AND (FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 66	00000460
	NXFUNC=LNKSPC(IADD)	00000470
	IF (AND (FLGSPC(NXFUNC),FL5MSK) .NE. 0.) GO TO 10	00000480
	NODSPC(IADD)=NODSPC(NXFUNC)	00000490
	LSTSPC(IADD)=LSTSPC(NXFUNC)	00000500
	LNKSPC(IADD)=LNKSPC(NXFUNC)	00000510
	FLGSPC(IADD)=FLGSPC(NXFUNC)	00000520
	RLOG=OR (FLGSPC(IADD),FL5MSK)	00000530
	FLGSPC(IADD)=ILOG	00000540
	IF (AND (FLGSPC(IADD),FL0MSK) .EQ. 0.) GO TO 9	00000550



	KVAL=LSTSPC(IADD)	00000560
8	KVAL=LSTSPC(KVAL)	00000570
	ISUB=LSTSPC(KVAL)	00000580
	IF (AND (FLGSPC (ISUB), FL0MSK) ,EQ. 0.) GO TO 8	00000590
	LSTSPC (KVAL)=IADD	00000600
9	IADD=NXFUNC	00000610
10	NODSPC (IADD)=NODSPC (REGASP)	00000620
		00000630
11	LSTSPC (IADD)=REGASP	00000640
	LNKSPC (IADD)=0	00000650
	FLGSPC (IADD)=0	00000660
	ISUB=LSTSPC (IADD)	00000670
	NODSPC (ISUB)=IADD	00000680
	ISUB=NODSPC (IADD)	00000690
	LSTSPC (ISUB)=IADD	00000700
	RETURN	00000710
72	IF (AND (FLGSPC (IADD), FL0MSK) ,NE. 0.) GO TO 20	00000720
	IF (IPOS,NE.1) GO TO 99	00000730
	IF (ITYP,EQ.3) GO TO 6	00000740
	RLOG=AND (FLGSPC (IADD), FLG67)	00000750
	ISTYP=ILOG	00000760
	IF (ISTYP,EQ.ITYP) GO TO 6	00000770
99	IVAL=-1	00000780
	RETURN	00000790
20	IND=0	00000800
	RLOG= OR (FLGSPC (IADD), FL4MSK)	00000810
	FLGSPC (IADD)=ILOG	00000820
	LAST=IADD	00000830
	IF (JPOS,121,99,21	00000840
121	ISUB=LSTSPC (IADD)	00000850
	LAST1=LNKSPC (ISUB)	00000860
	THIS=LAST1	00000870
	GO TO 27	00000880
21	IF (JPOS,LT.0) GO TO 80	00000890
	THIS=LSTSPC (LAST)	00000900
	IF (AND (FLGSPC (THIS), FL0MSK) ,NE. 0.) GO TO 99	00000910
	GO TO 27	00000920
80	THIS=LNKSPC (LAST)	00000930
	IF (THIS,EQ, LAST1) GO TO 99	00000940
27	IF (ITYP,EQ.3) GO TO 23	00000950
	RLOG=AND (FLGSPC (THIS), FLG67)	00000960
	ISTYP=ILOG	00000970
	IF (ISTYP,EQ.ITYP) GO TO 23	00000980
22	LAST=THIS	00000990
	GO TO 21	00010000
23	IND=IND+1	00010010
	IF (IND,NE,IPOS) GO TO 22	00010020
C	RETRIEVE THE IPOS <sup>TH</sup> OF THE KTYP <sup>TH</sup> VALUE BEFORE DELETING	00010030
	IVAL=NODSPC (THIS)	00010040
	MADD=IADD	00010050
	IF (JPOS,GT.0) GO TO 55	00010060
	NEXT=LNKSPC (THIS)	00010070
	IF (THIS,EQ, LAST1) GO TO 82	00010080
	LNKSPC (LAST)=NEXT	00010090
	GO TO 83	00011000



H2	ISUB=LSTSPC(IAUD)	00001110
	LNKSPC(ISUB)=NEXT	00001120
H3	IF(NEXT.EQ.LAST1) GO TO H4	00001130
	LSTSPC(NEXT)=LAST	00001140
	GO TO H5	00001150
H4	LSTSPC(IAUD)=LAST	00001160
H5	IAUD=THIS	00001170
	GO TO H6	00001180
55	NEXT=LSTSPC(THIS)	00001190
	IF(AND(FLGSPC(NEXT),FL0MSK) .NE. 0.) GO TO 50	00001200
	LNKSPC(NEXT)=LAST	00001210
	GO TO 24	00001220
50	ISUB=LSTSPC(IAUD)	00001230
	LNKSPC(ISUB)=LAST	00001240
24	IAUD=THIS	00001250
	JLAST=LNKSPC(THIS)	00001260
	ISUB=LSTSPC(JLAST)	00001270
	IF(AND(FLGSPC(ISUB),FL0MSK) .NE. 0.) LNKSPC(NEXT)=JLAST	00001280
	LSTSPC(LAST)=NEXT	00001290
H6	KLAST=LSTSPC(MADD)	00001300
	IF(LNKSPC(KLAST).NE.KLAST) GO TO 10	00001310
C		00001320
C	CONVERT TO SINGLE VALUE LIST	00001330
C		00001340
	LSTSPC(MADD)=NODSPC(KLAST)	00001350
	RLOG=AND(OR(FLGSPC(MADD),FLGSPC(KLAST)),NFLG02)	00001360
	FLGSPC(MADD)=ILOG	00001370
	FLGSPC(KLAST)=0	00001380
	LNKSPC(KLAST)=0	00001390
	NODSPC(KLAST)=NODSPC(REGASP)	00001400
	LSTSPC(KLAST)=REGASP	00001410
	ISUB=LSTSPC(KLAST)	00001420
	NODSPC(ISUB)=KLAST	00001430
	ISUB=NODSPC(KLAST)	00001440
	LSTSPC(ISUB)=KLAST	00001450
	GO TO 10	00001460
74	IF(AND(FLGSPC(IAUD),FL0MSK) .NE. 0.) GO TO 99	00001470
	IF(AND(FLGSPC(IAUD),FL2MSK) .EQ. 0.) GO TO 99	00001480
	LAST=LNKSPC(IAUD)	00001490
	NEXT=LSTSPC(IAUD)	00001500
	LSTSPC(LAST)=NEXT	00001510
	IF(AND(FLGSPC(NEXT),FL0MSK) .NE. 0.) GO TO 10	00001520
	LNKSPC(NEXT)=LAST	00001530
	GO TO 10	00001540
68	NEXT=LNKSPC(IAUD)	00001550
	NEXT1=NEXT	00001560
25	IF(LNKSPC(NEXT1).EQ.1ADD) GO TO 26	00001570
	NEXT1=LNKSPC(NEXT1)	00001580
	GO TO 25	00001590
26	LNKSPC(NEXT1)=NEXT	00001600
	KONFLC=1	00001610
	GO TO 10	00001620
	END	00001630



SUBROUTINE LVEXIT(N)	00000010
COMMON/LVARGS/LVFUNC,LVVAR,LVVAD,LVPOS,LVVTYP,LVVAL,	00000020
*LVHEAD,LVVNVL,LVUEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
COMMON/LVTABL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSGSP( 1)	00000040
COMMON /TYP/ NN(3),ERRFLG	00000050
COMMON /STRING/ NTYPE,NSTR	00000060
COMMON /NEED/ START,ASSOC,LEVEL,STOP	00000070
COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)	00000080
INTEGER R,RTEMP,STJ,STACK,ASSOC,START,STOP	00000090
COMMON /GIRL/ MM(19),OPRAND	00000100
COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ	00000110
COMMON /NTIMES/ NTIMES,MAXI	00000120
INTEGER STRING,HOL,ACTION,RIGHT,FUNC1,FUNC2,FUNC3,OPRAND	00000130
LOGICAL ERRFLG	00000140
C EXECUTE	00000150
GO TO 25000	00000160
25001 CONTINUE	00000170
IF(MAXJ,NE. 0) PRINT 100,MAXJ	00000180
100 FORMAT(1X,44H STATEMENT IS TOO COMPLEX. CORRECT TO CHAR. ,I3)	00000190
IF(MAXJ,EQ. 0) PRINT 200,MAXI,NSTR	00000200
200 FORMAT(1X,29H STATEMENT TOO LONG AT CHAR. ,I3,3H OF,I3)	00000210
C COMMENTS USED IN CASE OF GIRS PROBLEMS WHEN MEMORY USED UP	00000220
C GO TO 10	00000230
C IF(MAXJ,EQ. 0) GO TO 50	00000240
IF(MAXJ,EQ. 0) GO TO 10	00000250
DO 30 NCHAR=1,MAXJ	00000260
C STRING=HOL,NCHAR(-LEFT,-RIGHT,-HOL,-STRING)	00000270
C**** STRING * HOL	00000280
LVVPOS = NCHAR	00000290
LVVTYP = 3	00000300
LVFUNC= HOL	00000310
LVVARG= STRING	00000320
CALL LVFIND(LV2 A,LV2 B,LV2 C,LV2 D)	00000330
LV1 AAT = STRING	00000340
IF (LVVAL,NE,-1) LV1 AAT = LVVAL	00000350
LV1 AAX = LV1 AAT	00000360
C**** LV1 AAX - LEFT	00000370
LVVAD=-1	00000380
LVVTYP=-1	00000390
LVVPOS=1	00000400
LVFUNC= LEFT	00000410
LVVARG=LV1 AAX	00000420
CALL LVDELETE	00000430
LV1 AAX = LV1 AAT	00000440
C**** LV1 AAX - RIGHT	00000450
LVVAD=-1	00000460
LVVTYP=-1	00000470
LVVPOS=1	00000480
LVFUNC= RIGHT	00000490
LVVARG=LV1 AAX	00000500
CALL LVDELETE	00000510
LV1 AAX = LV1 AAT	00000520
C**** LV1 AAX - HOL	00000530
LVVAD=-1	00000540
LVVTYP=-1	00000550



LVVPOS=1			00000560
LVFUNC=	HOL		00000570
LVVARG=LV1	AAX		00000580
CALL LVULET			00000590
LV1	AAX = LV1	AAT	00000600
C****	LV1	AAX -	00000610
		STRING	00000620
LVVAD=-1			00000630
LVVTYP=-1			00000640
LVVPOS=1			00000650
LVFUNC=	STRING		00000660
LVVARG=LV1	AAX		00000670
CALL LVULET			00000680
30	CONTINUE		00000690
C	STRING-STRING		00000700
C****	STRING	-	00000710
		STRING	00000720
LVVAD=-1			00000730
LVVTYP=-1			00000740
LVVPOS=1			00000750
LVFUNC=	STRING		00000760
LVVARG=	STRING		00000770
CALL LVULET			00000780
C	OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)		00000790
LV1	AAT =	OPRAND	00000800
C****	LV1	AAT -	00000810
		OPRAND	00000820
LVVAD=-1			00000830
LVVTYP=-1			00000840
LVVPOS=1			00000850
LVFUNC=	OPRAND		00000860
LVVARG=LV1	AAT		00000870
CALL LVULET			00000880
C****	LV1	AAT -	00000890
		STRING	00000900
LVVAD=-1			00000910
LVVTYP=-1			00000920
LVVPOS=1			00000930
LVFUNC=	STRING		00000940
LVVARG=LV1	AAT		00000950
CALL LVULET			00000960
C****	LV1	AAT -	00000970
		ACTION	00000980
LVVAD=-1			00000990
LVVTYP=-1			00001000
LVVPOS=1			00001010
LVFUNC=	ACTION		00001020
LVVARG=LV1	AAT		00001030
CALL LVULET			00001040
C****	LV1	AAT -	00001050
		FUNC1	00001060
LVVAD=-1			00001070
LVVTYP=-1			00001080
LVVPOS=1			00001090
LVFUNC=	FUNC1		00001100
LVVARG=LV1	AAT		
CALL LVULET			
10	CONTINUE		
REWIND 19			
NTIMES=0			
C 10	CONTINUE		



C	REWIND 99	00001110
C	NTIMES=0	00001120
	J=NSTR+1	00001130
	R=STOP	00001140
	STJ=R	00001150
	ERRFLG=.TRUE.	00001160
	JSTACK=1	00001170
	STACK(JSTACK,1)=STOP	00001180
	STACK(JSTACK,2)=100	00001190
	STACK(JSTACK,3)=J	00001200
	STACK(JSTACK,4)=0	00001210
C	NSTR=0	00001220
	NSTR=MAXJ	00001230
C	COMPLETE	00001240
	RETURN	00001250
25000	CONTINUE	00001260
	LV2A=0	00001270
	LV2B=0	00001280
	LV2C=0	00001290
	LV2D=0	00001300
	LV2E=0	00001310
	LV2F=0	00001320
	LV2G=0	00001330
	LV2H=0	00001340
	GO TO 25001	00001350
	END	00001360

	SUBROUTINE LVFECH(N)	00000010
	INTEGER FLGSPC,SEQSPC,REGASP	00000020
	COMMON /LVTAHL/ MAPSZ,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)	00000030
	COMMON/LVVTR1/MEMSZ,REGASP,NODSPC( 1)/LVVTR2/LSTSPC( 1)/	00000040
	*LVVTR3/LNKSPC( 1)/LVVTR4/FLGSPC( 1)	00000050
	COMMON/LVRAND/KPRIME,KS,KX,KDY,KDX,KTEMP	00000060
	READ(N) MEMSZ,REGASP,KPRIME,KS,KX,KTEST,KDY,KTEMP,KDX,KNUM	00000070
	*,ISEQSZ	00000080
	READ(N) (NODSPC(I),I=1,MEMSZ)	00000090
	READ(N) (LSTSPC(I),I=1,MEMSZ)	00000100
	READ(N) (LNKSPC(I),I=1,MEMSZ)	00000110
	READ(N) (FLGSPC(I),I=1,MEMSZ)	00000120
	READ(N) (SEQSPC(I),I=1,ISEQSZ)	00000130
	PRINT 10	00000140
10	FORMAT(////1X,34H GRAPH HAS BEEN PLACED INTO MEMORY//)	00000150
	RETURN	00000160
	END	00000170



SUBROUTINE LVFIND(INDEX,INDXAD,KFUNC,KARG)	00000010
COMMON/LVARG5/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL,	00000020
* IDSTY,IVAL5(10),ITYP1(10),NSKIP	00000030
INTEGER FLGSPC,REGASP,FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,	00000040
* FLG67,SEQSPC	00000050
COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67	00000060
COMMON /LVTAHL/ MAPSIZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)	00000070
COMMON/LVVTR1/MEMSZE,REGASP,NODSPC( 1)/LVVTR2/LSTSPC( 1)/	00000080
*LVVTR3/LNKSPC( 1)/LVVTR4/FLGSPC( 1)	00000090
EQUIVALENCE(RLOG,ILOG)	00000100
DATA NFLAG4/247/	00000110
IADD=IFUNC*IARG	00000120
IF(IADD.GT,MEMSZE) IADD=IADD-MEMSZE	00000130
LSTHED=0	00000140
IF(AND(FLGSPC(IADD),FL5MSK) .EQ. 0.) GO TO 99	00000150
1 IF(NODSPC(IADD),EQ,IARG) GO TO 4	00000160
IADD=LNKSPC(IADD)	00000170
IF(AND(FLGSPC(IADD),FL5MSK) .NE. 0.) GO TO 99	00000180
GO TO 1	00000190
4 IF(AND(FLGSPC(IADD),FL0MSK) .NE. 0.) GO TO 14	00000200
RLOG=AND(FLGSPC(IADD),FLG67)	00000210
ISTYP=ILOG	00000220
IF(ITYP,EQ,3) GO TO 11	00000230
IF(ISTYP,EQ,3)ISTYP=2	00000240
IF(ISTYP.NE,ITYP) GO TO 99	00000250
11 IVAL=LSTSPC(IADD)	00000260
IF((IPOS.NE,1).AND,(IPOS.NE,-1)) GO TO 99	00000270
RLOG=AND(FLGSPC(IADD),FLG67)	00000280
ITYP=ILOG	00000290
LSTHED=-1	00000300
RETURN	00000310
14 LSTHED=IADD	00000320
IND=0	00000330
KINDEX=IABS(INDEX)	00000340
JPOS=IABS(IPOS)	00000350
IF(NSKIP,EQ,1) GO TO 50	00000360
IF((KFUNC.NE,IFUNC).OR,(KARG.NE,IARG)) GO TO 50	00000370
IF(AND(FLGSPC(LSTHED),FL4MSK) .NE. 0.) GO TO 50	00000380
IF((IPOS*INDEX) .LE. 0) GO TO 50	00000390
IF(JPOS.LT,2) GO TO 50	00000400
NDX=FLGSPC(INDXAD)	00000410
IF(AND(NDX,FL5MSK) .NE. 0.) GO TO 50	00000420
IF(AND(NDX,FL1MSK) .EQ. 0.) GO TO 50	00000430
IF(JPOS.GE,KINDEX) GO TO 25	00000440
IF((JPOS*JPOS).LE,KINDEX) GO TO 50	00000450
IF(KPOS)30,99,40	00000460
50 RLOG=AND(FLGSPC(LSTHED),NFLAG4)	00000470
FLGSPC(LSTHED)=ILOG	00000480
IF(IPOS) 20,99,10	00000490
C	00000500
COUNT DOWN FROM THE TOP OF THE LIST	00000510
C	00000520
10 IADD=LSTSPC(IADD)	00000530
IF(AND(FLGSPC(IADD),FL0MSK) .NE. 0.) GO TO 99	00000540
RLOG=AND(FLGSPC(IADD),FLG67)	00000550



ISTYP=ILOG	00000560
IF(ITYP.EQ.3) GO TO 22	00000570
IF(ISTYP.EQ.3) ISTYP=2	00000580
IF(ISTYP.NE.ITYP) GO TO 10	00000590
22 IND=IND+1	00000600
IF(IND.NE.JPOS) GO TO 10	00000610
28 IVAL=NODSPC(IADD)	00000620
RLOG=AND(FLGSPC(IADD),FLG67)	00000630
ITYP=ILOG	00000640
55 INDEX=KPOS	00000650
INDXAD=IADD	00000660
KFUNC=IFUNC	00000670
KARG=IARG	00000680
RETURN	00000690
C	00000700
COUNT UP FROM THE BOTTOM OF THE LIST	00000710
C	00000720
20 IADD=LSTSPC(IADD)	00000730
KTEST=0	00000740
23 IADD=LNKSPC(IADD)	00000750
IF(KTEST.EQ.0) GO TO 24	00000760
ISUB=LSTSPC(IADD)	00000770
IF(AND(FLGSPC(ISUB),FL0MSK).NE.0.) GO TO 99	00000780
24 KTEST=1	00000790
RLOG=AND(FLGSPC(IADD),FLG67)	00000800
ISTYP=ILOG	00000810
IF(ITYP.EQ.3) GO TO 21	00000820
IF(ISTYP.EQ.3) ISTYP=2	00000830
IF(ISTYP.NE.ITYP) GO TO 23	00000840
21 IND=IND+1	00000850
IF(IND.NE.JPOS) GO TO 23	00000860
29 IVAL=NODSPC(IADD)	00000870
RLOG=AND(FLGSPC(IADD),FLG67)	00000880
ITYP=ILOG	00000890
GO TO 55	00000900
25 IF(KPOS)40,99,30	00000910
C	00000920
COUNT DOWN FROM INDXADD	00000930
C	00000940
30 JPOS=IABS(JPOS-KINDEX)	00000950
IADD=INDXAD	00000960
IF(JPOS.EQ.0) GO TO 28	00000970
GO TO 10	00000980
C	00000990
COUNT UP FROM INDXADD	00001000
C	00001010
40 JPOS=IABS(JPOS-KINDEX)	00001020
IADD=INDXAD	00001030
IF(JPOS.EQ.0) GO TO 29	00001040
KTEST=1	00001050
GO TO 23	00001060
99 IVAL=-1	00001070
INDEX=0	00001080
INDXAD=0	00001090
KFUNC=0	00001100
KARG=0	00001110
RETURN	00001120
END	00001130



SUBROUTINE LVGRN(NODE)	00000010
INTEGER FLGSPC,REGASP	00000020
COMMON/LVVTR1/MEMSIZE,REGASP,NODSPC( 1)/LVVTR2/LSTSPC( 1)/	00000030
*LVVTR3/LNKSPC( 1)/LVVTR4/FLGSPC( 1)	00000040
COMMON/LVRAND/KPRIME,KSEED,NROW,KDNODE,KDROW,KTEMP	00000050
NODE=KTEMP+KDNODE	00000060
KTEMP=NODE	00000070
KDNODE=KDNODE+1	00000080
IF(NODE .GT. MEMSIZE) GO TO 5	00000090
RETURN	00000100
5 IF(NROW .GT. KPRIME) GO TO 10	00000110
NROW=NROW+KSEED	00000120
IF(NROW .GT. KPRIME) NROW=NROW-KPRIME	00000130
NODE=NROW	00000140
KTEMP=NODE	00000150
KDNODE=KPRIME+1	00000160
IF(NODE .NE. KSEED) RETURN	00000170
NROW=0	00000180
KDROW=KPRIME	00000190
10 KDROW=KDROW+1	00000200
NROW=NROW+KDROW	00000210
NODE=NROW	00000220
KTEMP=NODE	00000230
KDNODE=KDROW	00000240
IF(NODE .GT. MEMSIZE) GO TO 20	00000250
RETURN	00000260
20 PRINT 15	00000270
15 FORMAT(1H *1X,47H ERROR...NUMBER OF NODES EXCEEDS REQUEST MEMORY/	00000280
*27H THE PROGRAM IS TERMINATED.)	00000290
STOP	00000300
END	00000310



SUBROUTINE LVNSRT	00000010
COMMON/LVARG\$ /IFUNC,IARG,IADD,IPOS,ITYP2,IVAL,LSTHED,NVAL,	00000020
* IDSTRY,IVAL\$ (10),ITYP (10),NSKIP	00000030
INTEGER FLGSPC,FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FL67,REGASP,TEMP,THIS,	00000040
* FLGTMP,HEAD,OLDLOC,ASPREG,SEQSPC,FL3MSK,FL4MSK	00000050
COMMON/LVVTR1/MEM\$ZE,REGASP,NODSPC ( 1)/LVVTR2/LSTSPC ( 1)/	00000060
*LVVTR3/LNKSPC ( 1)/LVVTR4/FLGSPC ( 1)	00000070
COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FL67	00000080
COMMON /LVTABL/ MAP\$ZE,MAP (1) /LVVSEQ/ ISEQ\$7,SEQSPC (1)	00000090
EQUIVALENCE (ILOG,RLOG)	00000100
DATA NFLG67/252/	00000110
C	00000120
FLGTMP=FL1MSK	00000130
C	00000140
IF (REGASP,EQ,LSTSPC (REGASP)) GO TO 98	00000150
C	00000160
C FORM FIRST WORD OF SINGLE OR MULTIVALUED FUNCTION	00000170
IF (NVAL,EQ,1) GO TO 20	00000180
LSTTMP=REGASP	00000190
RLOG=OR (FL0MSK, OR (FLGTMP,FL2MSK))	00000200
FLGTMP=ILOG	00000210
GO TO 21	00000220
20 LSTTMP=IVAL\$ (1)	00000230
21 RLOG=OR (FLGTMP,ITYP (1))	00000240
FLGTMP=ILOG	00000250
C	00000260
C-----	00000270
C-DETERMINE ADDRESS FOR FUNCTION	00000280
IADD=IFUNC+IARG	00000290
IF (IADD,GT,MEM\$ZE) IADD=IADD-MEM\$ZE	00000300
C	00000310
C IF THAT ADDRESS IS ALREADY IN WORKING SPACE, GO TO 25	00000320
IF (IDSTRY=1) 125,300,350	00000330
125 IF (AND (FL1MSK,FLGSPC (IADD)) ,NE, 0.) GO TO 25	00000340
C	00000350
C UPDATE REGASP (IF NECESSARY)	00000360
23 IF (IADD,EQ,REGASP) REGASP=LSTSPC (IADD)	00000370
C	00000380
C UPDATE AVAILABLE SPACE	00000390
ISUB=NOUSPC (IADD)	00000400
LSTSPC (ISUB)=LSTSPC (IADD)	00000410
ISUB=LSTSPC (IADD)	00000420
NOUSPC (ISUB)=NOUSPC (IADD)	00000430
C	00000440
C INSERT FUNCTION	00000450
24 NOUSPC (IADD)=IARG	00000460
LSTSPC (IADD)=LSTTMP	00000470
LNKSPC (IADD)=IADD	00000480
RLOG=OR (FL5MSK,OR (FLGSPC (IADD),FLGTMP))	00000490
RLOG=OR (RLOG,FL4MSK)	00000500
FLGSPC (IADD)=ILOG	00000510
C	00000520
C INSERT ANY ADDITIONAL FUNCTION VALUES	00000530
HEAD=IADD	00000540
OLDLOC=IADD	00000550



	IF (NVAL.GT.1) GO TO 50	00000560
C		00000570
C	IF LAST CELL OF AVAILAHLE SPACE WAS USED, WRITE MESSAGE	00000580
	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 409	00000590
	IVAL=IAHS(IVALS(1))	00000600
	RETURN	00000610
C		00000620
C	IF THAT ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST, GO TO 41	00000630
	25 IF (AND (FLGSPC(IADD),FL5MSK) .GT. 0.) GO TO 41	00000640
C		00000650
C	IF THAT ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST, GO TO 35	00000660
	R1=AND (FL2MSK,FLGSPC(IADD))	00000670
	R2=AND (FL0MSK,FLGSPC(IADD))	00000680
	IF (R1 .GT. 0. .AND. R2 .EQ. 0.) GO TO 35	00000690
C		00000700
C	-----	00000710
C	THE ADDRESS CONTAINS A FUNCTION ON A CONFLICT LIST,BUT NOT THE HEAD OF	00000720
	THIS=IADD	00000730
C		00000740
C	FIND THE PRECEDING FUNCTION ON THE CONFLICT LIST	00000750
	26 ISUB=LNKSPC(THIS)	00000760
	IF (LNKSPC(ISUB) .EQ. IADD) GO TO 27	00000770
	THIS=LNKSPC(THIS)	00000780
	GO TO 26	00000790
	27 LAST=LNKSPC(THIS)	00000800
	NEWLOC=REGASP	00000810
	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 98	00000820
C		00000830
C	UPDATE AVAILAHLE SPACE AND REGASP	00000840
	28 ISUB=NOUSPC(REGASP)	00000850
	LSTSPC(ISUB)=LSTSPC(REGASP)	00000860
	ISUB=LSTSPC(REGASP)	00000870
	NOUSPC(ISUB)=NOUSPC(REGASP)	00000880
	REGASP=LSTSPC(REGASP)	00000890
		00000900
C	MOVE THE FUNCTION ON A CONFLICT LIST TO THE FIRST CELL OF AVAILABLE	00000910
	29 NOUSPC(NEWLOC)=NOUSPC(IADD)	00000920
	LSTSPC(NEWLOC)=LSTSPC(IADD)	00000930
	LNKSPC(NEWLOC)=LNKSPC(IADD)	00000940
	FLGSPC(NEWLOC)=FLGSPC(IADD)	00000950
	FLGSPC(IADD)=0	00000960
	LNKSPC(LAST)=NEWLOC	00000970
C		00000980
C	INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST	00000990
	NOUSPC(IADD)=IARG	00001000
	LNKSPC(IADD)=IADD	00001010
	LSTSPC(IADD)=LSTMP	00001020
	RLOG=OR (FL5MSK,OR (FLGSPC(IADD),FLGTMP))	00001030
	RLOG=OR (RLOG,FL4MSK)	00001040
	FLGSPC(IADD)=ILOG	00001050
	IF (AND (FLGSPC(NEWLOC),FL0MSK) .EQ. 0.) GO TO 34	00001060
C		00001070
C	IF THE FUNCTION THAT WAS MOVED IS THE HEAD OF A MULTIVALUE LIST, FIX	00001080
	NEXT=LSTSPC(NEWLOC)	00001090
	30 NEXT=LSTSPC(NEXT)	00001100



IF (LSTSPC(NEXT).NE.IADD) GO TO 30	00001110
LSTSPC(NEXT)=NEWLOC	00001120
C	00001130
C INSERT ANY ADDITIONAL FUNCTION VALUES	00001140
34 HEAD=IADD	00001150
OLDLOC=IADD	00001160
IF (NVAL.GT.1) GO TO 50	00001170
IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 909	00001180
IVAL=IABS(IVAL(1))	00001190
RETURN	00001200
C	00001210
C-----	00001220
C- THE ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST	00001230
35 NEWLOC=REGASP	00001240
IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 98	00001250
C	00001260
C UPDATE AVAILABLE SPACE AND REGASP	00001270
36 ISUB=NODSPC(REGASP)	00001280
LSTSPC(ISUB)=LSTSPC(REGASP)	00001290
ISUB=LSTSPC(REGASP)	00001300
NODSPC(ISUB)=NODSPC(REGASP)	00001310
REGASP=LSTSPC(REGASP)	00001320
C	00001330
C MOVE THE VALUE ON A MULTIVALUE LIST TO THE FIRST CELL OF AVAILABLE	00001340
37 NODSPC(NEWLOC)=NODSPC(IADD)	00001350
LSTSPC(NEWLOC)=LSTSPC(IADD)	00001360
LNKSPC(NEWLOC)=LNKSPC(IADD)	00001370
FLGSPC(NEWLOC)=FLGSPC(IADD)	00001380
FLGSPC(IADD)=0	00001390
C	00001400
C RESET POINTERS	00001410
C	00001420
L1=LSTSPC(NEWLOC)	00001430
IF (AND (FL0MSK,FLGSPC(L1)) .EQ. 0.) GO TO 200	00001440
ISUB=LSTSPC(L1)	00001450
LNKSPC(ISUB)=NEWLOC	00001460
GO TO 201	00001470
200 LNKSPC(L1)=NEWLOC	00001480
201 ISUB=LNKSPC(NEWLOC)	00001490
KZVAL=LSTSPC(ISUB)	00001500
IF (AND (FLGSPC(KZVAL),FL0MSK) .NE. 0.) GO TO 38	00001510
ISUB=LNKSPC(NEWLOC)	00001520
LSTSPC(ISUB)=NEWLOC	00001530
GO TO 39	00001540
38 LSTSPC(KZVAL)=NEWLOC	00001550
39 NODSPC(IADD)=IARG	00001560
C INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST	00001570
LNKSPC(IADD)=IADD	00001580
LSTSPC(IADD)=LSTTMP	00001590
RLOG=OR (FL5MSK,OR (FLGSPC(IADD),FLGTMP))	00001600
RLOG=OR (RLOG,FL4MSK)	00001610
FLGSPC(IADD)=ILOG	00001620
IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 909	00001630
IVAL=IABS(IVAL(1))	00001640
RETURN	00001650



C	00001660
C-----	00001670
C-THE ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST	00001680
41 THIS=IADD	00001690
C	00001700
C IF THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST, GO TO 60	00001710
42 IF (NODSPC (THIS),EQ,IARG) GO TO 43	00001720
ISUB=LNKSPC (THIS)	00001730
IF (AND (FLGSPC (ISUB),FL5MSK) .NE. 0.) GO TO 60	00001740
THIS=LNKSPC (THIS)	00001750
GO TO 42	00001760
C	00001770
C-----	00001780
C-THE FUNCTION TO BE INSERTED IS ON THE CONFLICT LIST	00001790
43 HEAD=THIS	00001800
IF (AND (FLGSPC (THIS),FL0MSK) .EQ. 0.) GO TO 51	00001810
NEXT=LSTSPC (THIS)	00001820
C	00001830
C OLDLOC IS THE LOCATION OF THE LAST VALUE ON THE MULTIVALUE LIST	00001840
C	00001850
OLDLOC=LNKSPC (NEXT)	00001860
C	00001870
C-----	00001880
C-INSERT ADDITIONAL FUNCTION VALUES	00001890
50 LSTASP=NODSPC (REGASP)	00001900
IN=0	00001910
GO TO 56	00001920
C	00001930
C-----	00001940
C-FORM MULTIVALUE LIST TO ADD VALUE(S) TO SINGLE-VALUED FUNCTION	00001950
51 IN=0	00001960
IF (REGASP,EQ,LSTSPC (REGASP)) GO TO 98	00001970
LSTASP=NODSPC (REGASP)	00001980
NEWLOC=REGASP	00001990
REGASP=LSTSPC (REGASP)	00002000
NODSPC (NEWLOC)=LSTSPC (THIS)	00002010
RLOG=OR (FLGSPC (NEWLOC),AND (FLGSPC (THIS),FLG67))	00002020
FLGSPC (NEWLOC)=ILOG	00002030
RLOG=OR (FL0MSK,OR (FL2MSK,AND (NFLG67,FLGSPC (THIS))))	00002040
FLGSPC (THIS)=ILOG	00002050
OLDLOC=THIS	00002060
C	00002070
C-----	00002080
C INSERT ANOTHER VALUE ON MULTIVALUE LIST	00002090
52 RLOG=OR (FL1MSK,OR (FL2MSK,FLGSPC (NEWLOC)))	00002100
FLGSPC (NEWLOC)=ILOG	00002110
LSTSPC (OLDLOC)=NEWLOC	00002120
LNKSPC (NEWLOC)=OLDLOC	00002130
55 OLDLOC=NEWLOC	00002140
56 NEWLOC=REGASP	00002150
IF (IN,GT,0) GO TO 57	00002160
C	00002170
C NO VALUES HAVE BEEN INSERTED YET	00002180
IN=1	00002190
GO TO 58	00002200



C		00002210
C	SOME VALUES HAVE BEEN INSERTED	00002220
57	IF (IN.EQ.NVAL) GO TO 67	00002230
	IN=IN+1	00002240
C		00002250
58	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 909	00002260
581	REGASP=LSTSPC(REGASP)	00002270
582	NODSPC(NEWLOC)=IVAL5(IN)	00002280
	RLOG=OR(ITYP(IN),FLGSPC(NEWLOC))	00002290
	FLGSPC(NEWLOC)=ILOG	00002300
	GO TO 52	00002310
C		00002320
C	END MULTIVALUE LIST AND UPDATE AVAILABLE SPACE	00002330
67	LSTSPC(OLDLOC)=HEAD	00002340
	NODSPC(REGASP)=LSTASP	00002350
	LSTSPC(LSTASP)=REGASP	00002360
	IVAL=IABS(IVAL5(1))	00002370
	ISUB=LSTSPC(HEAD)	00002380
	LNKSPC(ISUB)=OLDLOC	00002390
	NVAL=IN	00002400
	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 909	00002410
	RETURN	00002420
C		00002430
C	-----	00002440
C	THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST	00002450
60	ASPREG=REGASP	00002460
	LSTASP=NODSPC(REGASP)	00002470
	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 98	00002480
C		00002490
C	UPDATE AVAILABLE SPACE AND REGASP	00002500
601	ISUB=NODSPC(REGASP)	00002510
	LSTSPC(ISUB)=LSTSPC(REGASP)	00002520
	ISUB=LSTSPC(REGASP)	00002530
	NODSPC(ISUB)=NODSPC(REGASP)	00002540
	REGASP=LSTSPC(REGASP)	00002550
C		00002560
C	INSERT FUNCTION IN FIRST CELL OF AVAILABLE SPACE	00002570
61	NODSPC(ASPREG)=IARG	00002580
	IF (NVAL.EQ.1) GO TO 611	00002590
	LSTSPC(ASPREG)=REGASP	00002600
	RLOG=OR(FL0MSK,OR(FL2MSK,FLGSPC(ASPREG)))	00002610
	FLGSPC(ASPREG)=ILOG	00002620
	GO TO 612	00002630
611	LSTSPC(ASPREG)=IVAL5(1)	00002640
612	RLOG=OR(FL1MSK,OR(ITYP(1),FLGSPC(ASPREG)))	00002650
	RLOG=OR(RLOG,FL4MSK)	00002660
	FLGSPC(ASPREG)=ILOG	00002670
	LNKSPC(ASPREG)=LNKSPC(THIS)	00002680
	LNKSPC(THIS)=ASPREG	00002690
	IF (NVAL.EQ.1) GO TO 613	00002700
C		00002710
C	INSERT ADDITIONAL VALUES	00002720
	LSTASP=NODSPC(REGASP)	00002730
	OLDLOC=ASPREG	00002740
	HEAD=ASPREG	00002750



	IN=0	00002760
	GO TO 56	00002770
613	IF (REGASP.EQ.LSTSPC(REGASP)) GO TO 909	00002780
	IVAL=IARS(IVAL5(1))	00002790
	RETURN	00002800
C		00002810
C	DESTRUCTIVE INSERTION	00002820
C		00002830
350	IADD1=IADD	00002840
	INDEX=0	00002850
	CALL LVFIND(INDEX,INDEX,INDEX,INDEX)	00002860
	RLOG=OR(FLGSPC(IADD),FL4MSK)	00002870
	FLGSPC(IADD)=ILOG	00002880
	IF (IVAL.EQ.-1) GO TO 90	00002890
	IF (LSTHED) 354,40,356	00002900
354	LSTSPC(IADD)=IVAL5(1)	00002910
	GO TO 365	00002920
356	NODSPC(IADD)=IVAL5(1)	00002930
365	RLOG=OR(ITYP(1),AND(NFL667,FLGSPC(IADD)))	00002940
	FLGSPC(IADD)=ILOG	00002950
	GO TO 360	00002960
90	IF (IPOS) 91,99,92	00002970
91	IPOS=IPOS+1	00002980
	GO TO 93	00002990
92	IPOS=IPOS-1	00003000
93	IADD=IADD1	00003010
	IF (IPOS.EQ.0) GO TO 125	00003020
	INDEX=0	00003030
	CALL LVFIND(INDEX,INDEX,INDEX,INDEX)	00003040
	IF (IVAL.EQ.-1) GO TO 99	00003050
	IF (IPOS.LT. 0) GO TO 370	00003060
	IADD=IADD1	00003070
	GO TO 125	00003080
370	NEWLOC=REGASP	00003090
	IF (LSTHED) 325,99,375	00003100
C	UPDATE AVAILAHLE SPACE	00003110
375	ISUB=NODSPC(REGASP)	00003120
	LSTSPC(ISUB)=LSTSPC(REGASP)	00003130
	ISUB=LSTSPC(REGASP)	00003140
	NODSPC(ISUB)=NODSPC(REGASP)	00003150
	REGASP=LSTSPC(REGASP)	00003160
	GO TO 377	00003170
C		00003180
C	NONDESTRUCTIVE INSERTION	00003190
C		00003200
300	IADD1=IADD	00003210
	NEWLOC=REGASP	00003220
	INDEX=0	00003230
	CALL LVFIND(INDEX,INDEX,INDEX,INDEX)	00003240
	RLOG=OR(FLGSPC(IADD),FL4MSK)	00003250
	FLGSPC(IADD)=ILOG	00003260
	IF (IVAL.EQ.-1) GO TO 90	00003270
	IF (LSTHED) 344,90,346	00003280
344	IF (IPOS.GT. 0) GO TO 325	00003290
	IADD=IADD1	00003300



AD-A043 925

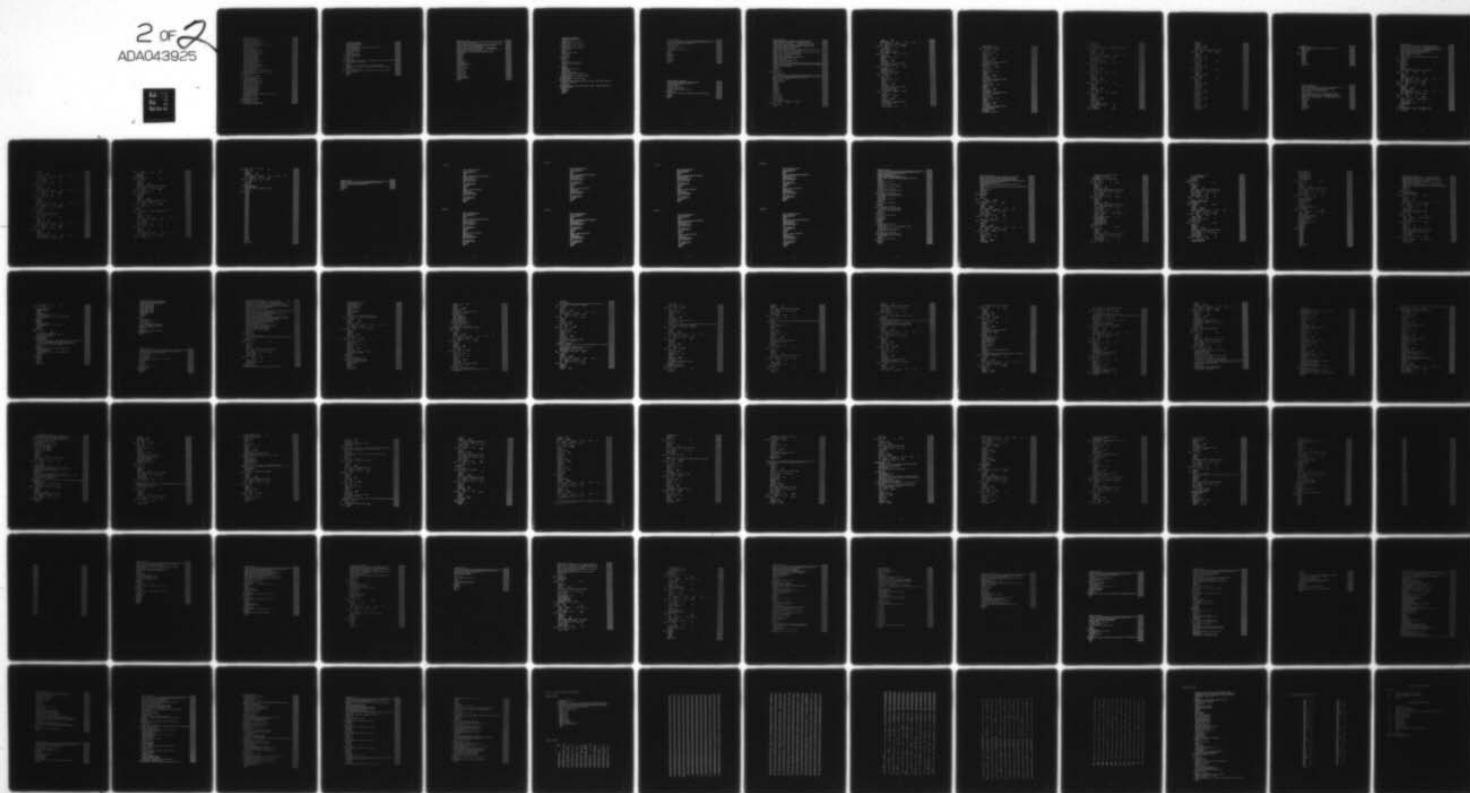
DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2  
MAINTENANCE MANUAL FOR AUDIT. A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)  
AUG 77 R J WYBRANIEC, R REGEN

UNCLASSIFIED

DTNSRDC-77-0075-VOL-4

NL

2 OF 2  
ADA043925



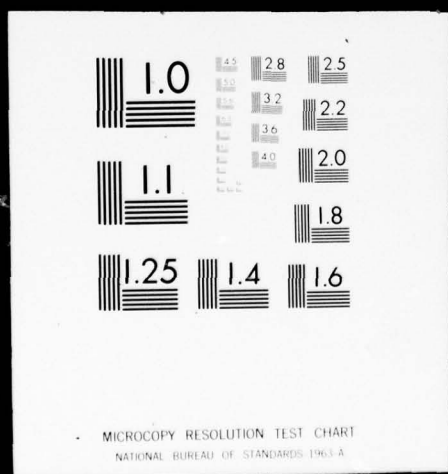
END  
DATE  
FILMED  
10-77  
DDC



2 OF 2

2

ADA043925





	GO TO 125	00003310
C	CREATE MULTIVALUE LIST	00003320
325	ISUB=NODSPC(REGASP)	00003330
	LSTSPC(ISUB)=LSTSPC(REGASP)	00003340
	ISUB=LSTSPC(REGASP)	00003350
	NODSPC(ISUB)=NODSPC(REGASP)	00003360
	REGASP=LSTSPC(REGASP)	00003370
	IF (REGASP, EQ, LSTSPC(REGASP)) GO TO 909	00003380
	NWLOC2=REGASP	00003390
C	UPDATE AVAILABLE SPACE	00003400
	ISUB=NODSPC(REGASP)	00003410
	LSTSPC(ISUB)=LSTSPC(REGASP)	00003420
	ISUB=LSTSPC(REGASP)	00003430
	NODSPC(ISUB)=NODSPC(REGASP)	00003440
	REGASP=LSTSPC(REGASP)	00003450
	NODSPC(NEWLOC)=IVALS(1)	00003460
	LSTSPC(NEWLOC)=NWLOC2	00003470
	LNKSPC(NEWLOC)=NWLOC2	00003480
	RLOG=OR (FLGTMP, FL2MSK)	00003490
	FLGSPC(NEWLOC)=ILOG	00003500
	NODSPC(NWLOC2)=LSTSPC(IADD)	00003510
	LSTSPC(NWLOC2)=IADD	00003520
	LNKSPC(NWLOC2)=NEWLOC	00003530
	R1=AND (FLGSPC(IADD), FLG67)	00003540
	RLOG=OR (R1, OR (FL1MSK, FL2MSK))	00003550
	FLGSPC(NWLOC2)=ILOG	00003560
	LSTSPC(IADD)=NEWLOC	00003570
	RLOG=OR (FL2MSK, OR (FL0MSK, FLGSPC(IADD)))	00003580
	FLGSPC(IADD)=ILOG	00003590
320	IF (REGASP, EQ, LSTSPC(REGASP)) GO TO 909	00003600
360	IVAL=IABS(IVALS(1))	00003610
	RETURN	00003620
C	UPDATE AVAILABLE SPACE	00003630
346	ISUB=NODSPC(REGASP)	00003640
	LSTSPC(ISUB)=LSTSPC(REGASP)	00003650
	ISUB=LSTSPC(REGASP)	00003660
	NODSPC(ISUB)=NODSPC(REGASP)	00003670
	REGASP=LSTSPC(REGASP)	00003680
	IF (IPOS, LT, 0) GO TO 347	00003690
377	ISTLOC=LNKSPC(IADD)	00003700
	NODSPC(NEWLOC)=IVALS(1)	00003710
	LSTSPC(NEWLOC)=IADD	00003720
	LNKSPC(NEWLOC)=ISTLOC	00003730
	RLOG=OR (FLGTMP, FL2MSK)	00003740
	FLGSPC(NEWLOC)=ILOG	00003750
	ISUB=LSTSPC(ISTLOC)	00003760
	IF (AND (FLGSPC(ISUB), FL0MSK), EQ, 0.) GO TO 321	00003770
	ISUB=LSTSPC(ISTLOC)	00003780
	LSTSPC(ISUB)=NEWLOC	00003790
	GO TO 322	00003800
321	LSTSPC(ISTLOC)=NEWLOC	00003810
322	LNKSPC(IADD)=NEWLOC	00003820
	GO TO 320	00003830
347	NODSPC(NEWLOC)=IVALS(1)	00003840
	LSTSPC(NEWLOC)=LSTSPC(IADD)	00003850



LNKSPC(NEWLOC)=IADD	00003860
RLOG=OR(FLGTMP,FL2MSK)	00003870
FLGSPC(NEWLOC)=ILOG	00003880
ISUB=LSTSPC(IADD)	00003890
IF(AND(FLGSPC(ISUB),FL0MSK),EQ. 0.) GO TO 323	00003900
KZVAL=LSTSPC(IADD)	00003910
ISUB=LSTSPC(KZVAL)	00003920
LNKSPC(ISUB)=NEWLOC	00003930
GO TO 324	00003940
323 ISUB=LSTSPC(IADD)	00003950
LNKSPC(ISUB)=NEWLOC	00003960
324 LSTSPC(IADD)=NEWLOC	00003970
GO TO 320	00003980
98 IVAL=-3	00003990
PRINT 20001	00004000
20001 FORMAT(1X,78H ERROR...THERE IS NO ADDITIONAL SPACE FOR THE GRAPH,	00004010
*THE PROGRAM IS TERMINATED)	00004020
STOP	00004030
99 IVAL=-1	00004040
2 FORMAT(6H ONLY ,I4,28H VALUE(S) HAVE BEEN INSERTED)	00004050
22 FORMAT(1X,I5,JH(,I5,35H) USED LAST CELL OF AVAILAHLE SPACE)	00004060
RETURN	00004070
909 PRINT 22,IFUNC,IARG	00004080
C THIS INSERTION HAS FILLED GIRS MEMORY - CALL A USER SUPPLIED	00004090
C PROGRAM - LVEXIT.	00004100
IVAL=-1	00004110
RETURN	00004120
END	00004130



	SUBROUTINE LVSETP	00000010
	INTEGER FLGSPC,FLAGSP,REGASP,BINFIL,FL0MSK,FL1MSK,FL2MSK,FL5MSK,	00000020
	* FL3MSK,FL4MSK,FLG67,SEQSPC	00000030
	COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67	00000040
	COMMON/LVVTR5/HINFIL,KOMPAN,NODESP(1)/LVVTR6/LISTSP(1)	00000050
	* /LVVTR7/LINKSP(1)/LVVTR8/FLAGSP(1)	00000060
	COMMON /LVTAHL/ MAPSZ,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)	00000070
	COMMON/LVVTR1/MEMSZ,REGASP,NODSPC( 1)/LVVTR2/LSTSPC( 1)/	00000080
	*LVVTR3/LNKSPC( 1)/LVVTR4/FLGSPC( 1)	00000090
	COMMON/LVRAND/KPRIME,KSEED,NROW,KDNODE,KDROW,KTEMP	00000100
	FL0MSK=128	00000110
	FL1MSK=64	00000120
	FL2MSK=32	00000130
	FL5MSK=4	00000140
	FLG67=3	00000150
	FL3MSK=16	00000160
	FL4MSK=8	00000170
	KSEED=KPRIME/2	00000180
	NROW=KSEED	00000190
	KTEMP=KSEED-KPRIME	00000200
	KDNODE=KPRIME	00000210
	REGASP=1	00000220
	DO 10 I=2,MEMSZ	00000230
	LNKSPC(I)=0	00000240
	FLGSPC(I)=0	00000250
	NODSPC(I)=I-1	00000260
10	LSTSPC(I-1)=I	00000270
	FLGSPC(I)=0	00000280
	LNKSPC(I)=0	00000290
	NODSPC(1)=MEMSZ	00000300
	LSTSPC(MEMSZ)=1	00000310
	RETURN	00000320
	END	00000330



```

SUBROUTINE MODID(MODE)
  DIMENSION IBUF(80),IEND(3)
  IF(MODE .NE. 0) GO TO 5
  WRITE(13,1)
1  FORMAT(5X,16H OUTPUT DEVICE X)
  WRITE(14,2)
2  FORMAT(5X,16H OUTPUT DEVICE Y)
  WRITE(15,3)
3  FORMAT(5X,16H OUTPUT DEVICE Z)
5  DO 10 I=10,12
    ENDFILE I
    REWIND I
    READ(I,7,END=8) ICHAR
7  FORMAT(A1)
    IEND(I-9)=0
    GO TO 10
8  IEND(I-9)=1
10  CONTINUE
    IEOF=0
    DO 15 I=1,3
      IF(IEND(I) .NE. 0) GO TO 15
      IF(IEOF .EQ. 1) GO TO 40
      IEOF=1
      IOUT=9+I
15  CONTINUE
      IF(IEOF .EQ. 0) GO TO 50
      REWIND IOUT
      IOUT2=IOUT+3
      WRITE(IOUT2,23) MODE
20  FORMAT(//20X,12H MODE INDEX=,I3)
      DO 30 I=1,100
        READ(IOUT,25,END=61) (IBUF(J),J=1,80)
25  FORMAT(80A1)
30  WRITE(IOUT2,25) (IBUF(J),J=1,80)
40  WRITE(6,45)
45  FORMAT(5X,73H***** ERROR IN ROLL CALL CHECK - MORE THAN 1 OUTPUT D
      *EVICE WAS WRITTEN ON)
      GO TO 60
50  WRITE(6,55)
55  FORMAT(5X,66H***** ERROR IN ROLL CALL CHECK - NO OUTPUT DEVICES WE
      *RE WRITTEN ON)
60  REWIND 10
    REWIND 11
    REWIND 12
    RETURN
  END

```



FUNCTION NEXT(IA)	00000010
COMMON A(1326),U(500),IDTHL(11,500),INITIU(3),LASTIU(3),ISWCH(3),	00000020
* NXTIU(2),JPTH,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
INTEGER A,HLANK	00000050
DATA HLANK/1H /	00000060
IF (IA .GT. N) GO TO 15	00000070
DO 10 I=IA,N	00000080
IF (A(I) .EQ. HLANK) GO TO 10	00000090
NEXT=A(I)	00000100
IA=I+1	00000110
RETURN	00000120
10 CONTINUE	00000130
NEXT=HLANK	00000140
IA=N+1	00000150
RETURN	00000160
15 NEXT=HLANK	00000170
IA=IA+1	00000180
RETURN	00000190
END	00000200

FUNCTION NXTHLK(ILOC,IEND)	00000010
COMMON/HASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000020
COMMON/LABELS/STATRA(2,200),NLABEL	00000030
INTEGER STATRA,HITGET	00000040
I=IBLOCK(ILOC)	00000050
IF (I .EQ. 999) GO TO 10	00000060
IF (I .EQ. 998) GO TO 5	00000070
NXTHLK=HITGET(STATRA(2,I),32,14)	00000080
RETURN	00000090
5 NXTHLK=IEND+1	00000100
IF (NXTHLK .GT. NBLOCK) CALL ERROR(38,IDM1,IDM2,IDM3,IDM4)	00000110
RETURN	00000120
10 NXTHLK=0	00000130
RETURN	00000140
END	00000150



```

SUBROUTINE PARSE                                00000010
COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTP,LVVAL, 00000020
*LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP 00000030
COMMON/LVTAHL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSQSP( 1) 00000040
COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4), 00000050
COMMON/FUNC/ NARY(5,22),MARGS,IARGS(50),FNCLC(5),NFUNC 00000060
COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG 00000070
COMMON /STRING/ NTYPE,NSTR,STR 00000080
COMMON /GJWL/NTERMS,PLUS,MINUS,SLASH,LPAR,PPAR,COMMA,STAR,EXP,LT, 00000090
*LE,GT,GE,EQ,NE,OR,AND,NOT,EQUALS,OPRAND 00000100
COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ 00000110
COMMON /NEED/ START,ASSOC,LEVEL,STOP 00000120
COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG 00000130
COMMON /NTIMES/ NTIMES,I 00000140
COMMON/VAR/VFOR(30),NUMCHR,NCHRP,CHAR,NDICT 00000150
INTEGER TYPE1,TYPE2,START,TYP(3) 00000160
LOGICAL ERRFLG,FAIL 00000170
INTEGER STR(500),STEMP,ST,DICT(19) 00000180
EQUIVALENCE(DICT(1),PLUS) 00000190
INTEGER PLUS,MINUS,SLASH,LPAR,PPAR,COMMA,STAR,EXP,LT,LE,GT,GE,EQ, 00000200
*NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT 00000210
*,STRING,FUNC1,FUNC2,FUNC3 00000220
IF(NTIMES.GT. 0) GO TO 3 00000240
NTIMES=1 00000250
C EXECUTE 00000260
GO TO 25000 00000270
25001 CONTINUE 00000280
CALL PHONEY 00000290
CALL LVFECH(19) 00000300
READ(19)PLUS,MINUS,SLASH,LPAR,PPAR,COMMA,STAR,EXP,LT,LE,GT,GE,FO, 00000310
*NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT 00000320
*,STRING,FUNC1,FUNC2,FUNC3,NTERMS,(TYP(I),I=1,3) 00000330
3 IF(NSTR.LE. 0) RETURN 00000340
ERRFLG=.FALSE. 00000350
START=TYP(NTYPE) 00000360
NARGS=0 00000370
MAXJ=0 00000380
NOPAR=0 00000390
TYPE1=-1 00000400
TYPE2=-1 00000410
NARRAY=-1 00000420
NDEPTH=0 00000430
NDEP=0 00000440
NFLAG=0 00000450
DO 20 I=1,50 00000460
20 IARGS(I)=0 00000470
DO 22 J=1,22 00000480
DO 22 I=1,5 00000490
22 NARY(I,J)=0 00000500
DO 10 I=1,NSTR 00000510
C STRING(=HOL,I "NTEMP//4,HOL $ "NTEMP) 00000520
LV1 AAB = STRING 00000530
C*** LV1 AAB * HOL 00000540
LVVPOS = I 00000550
LVVTYP = 3 00000560

```



	LVFUNC=	HOL				00000570
	LVVARG=	LV1	AAB			00000580
	CALL LVFIND(LV2		A.LV2	B.LV2	C.LV2	00000590
	LV1	AAC = LV1	AAB			00000600
	IF (LVVAL.NE.-1)	LV1	AAC = LVVAL			00000610
C****	LV1	AAC	"	NTMP		00000620
	NTMP = LV1	AAC				00000630
	LVVTR = LVVAL					00000640
	LVVAL = -100					00000650
	IF (LVVTR.NE.-1)	GO TO		4		00000660
	CALL LVGRN(LV1	AAC)				00000670
C****	LV1	AAB	HOL	LV1	AAC	00000680
	LVDEST=	0				00000690
	LVTYP(1) =	0				00000700
	LVVALS(1) =	LV1	AAC			00000710
	LVVNVL =	1				00000720
	LVFUNC =	HOL				00000730
	LVVARG=LV1	AAB				00000740
	CALL LVNSRT					00000750
	IF (LVVAL.LT.0)	CALL LVEXIT(LVVAL)				00000760
	IF (LVVAL.LT.0)	RETURN				00000770
C****	LV1	AAC	"	NTMP		00000780
	NTMP = LV1	AAC				00000790
	4 CONTINUE					00000800
	IF (ERRFLG)	GO TO 25				00000810
	ST=IAHS(STR(I))					00000820
	IF (STR(I).LT.0)	GO TO 6				00000830
C	NTMP	HOL	""ST""			00000840
C****	NTMP	HOL	""			00000850
	LVDEST=	0				00000860
	LV1	AAB = ST				00000870
	LVTYP(1) =	1				00000880
	LVVALS(1) =	LV1	AAB			00000890
	LVDEST=	0				00000900
	LVVNVL =	1				00000910
	LVFUNC =	HOL				00000920
	LVVARG=	NTMP				00000930
	CALL LVNSRT					00000940
	IF (LVVAL.LT.0)	CALL LVEXIT(LVVAL)				00000950
	IF (LVVAL.LT.0)	RETURN				00000960
	IF (ERRFLG)	GO TO 25				00000970
C	STRING	STRING	OPRND//10			00000980
C****	STRING	STRING	OPRND			00000990
	LVDEST=	0				00010000
	LVTYP(1) =	0				00010010
	LVVALS(1) =	OPRND				00010020
	LVVNVL =	1				00010030
	LVFUNC =	STRING				00010040
	LVVARG=	STRING				00010050
	CALL LVNSRT					00010060
	IF (LVVAL.LT.0)	CALL LVEXIT(LVVAL)				00010070
	IF (LVVAL.LT.0)	RETURN				00010080
	LVVTR = LVVAL					00010090
	LVVAL = -100					00010100
	IF (LVVTR.NE.-1)	GO TO		10		00010110



IF (ERRFLG) GO TO 25	00001120
6 STEMP=DICT(1ST)	00001130
C STRING STRING ""STEMP""	00001140
C**** STRING STRING ""	00001150
LVDEST= 0	00001160
LV1 AAC = STEMP	00001170
LVTYPE(1) = 1	00001180
LVVALS(1) = LV1 AAC	00001190
LVDEST= 0	00001200
LVVNVL = 1	00001210
LVFUNC = STRING	00001220
LVVARG= STRING	00001230
CALL LVNSRT	00001240
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00001250
IF (LVVAL.LT.0) RETURN	00001260
IF (ERRFLG) GO TO 25	00001270
10 CONTINUE	00001280
LV1 AAD = OPRAND	00001290
LVDEST= 0	00001300
LV1 AAE = 0	00001310
LVTYPE(1) = 1	00001320
LVVALS(1) = LV1 AAE	00001330
LVDEST= 0	00001340
LVVNVL = 1	00001350
LVFUNC = FUNC2	00001360
LVVARG=LV1 AAD	00001370
CALL LVNSRT	00001380
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00001390
IF (LVVAL.LT.0) RETURN	00001400
LVDEST= 0	00001410
LV1 AAF = 0	00001420
LVTYPE(1) = 1	00001430
LVVALS(1) = LV1 AAF	00001440
LVDEST= 0	00001450
LVVNVL = 1	00001460
LVFUNC = FUNC3	00001470
LVVARG=LV1 AAD	00001480
CALL LVNSRT	00001490
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00001500
IF (LVVAL.LT.0) RETURN	00001510
LVDEST= 0	00001520
LV1 AAG = 0	00001530
LVTYPE(1) = 1	00001540
LVVALS(1) = LV1 AAG	00001550
LVDEST= 0	00001560
LVVNVL = 1	00001570
LVFUNC = LEVEL	00001580
LVVARG=LV1 AAD	00001590
CALL LVNSRT	00001600
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00001610
IF (LVVAL.LT.0) RETURN	00001620
CALL RECOG(FAIL)	00001630
IF (FAIL) GO TO 40	00001640
25 CONTINUE	00001650
IF (ERRFLG) PRINT 100	00001660



100	FORMAT(/)		00001670
	CALL PRN15		00001680
	NCHAR=0		00001690
30	NCHAR=NCHAR+1		00001700
C	STRING=HOL,NCHAR/35(-LEFT,-RIGHT,-HOL,-STRING/30/30)		00001710
C****	STRING * HOL		00001720
	LVVPOS = NCHAR		00001730
	LVVTYP = 3		00001740
	LVFUNC= HOL		00001750
	LVVARG= STRING		00001760
	CALL LVFINU(LV2 E,LV2 F,LV2 G,LV2 H)		00001770
	LV1 AAD = STRING		00001780
	IF (LVVAL,NE,-1) LV1 AAD = LVVAL		00001790
	LVVTR = LVVAL		00001800
	LVVAL = -100		00001810
	IF (LVVTR,EQ,-1) GO TO 35		00001820
	LV1 AAE = LV1 AAD		00001830
C****	LV1 AAE - LEFT		00001840
	LVVAD=-1		00001850
	LVVTYP=-1		00001860
	LVVPOS=1		00001870
	LVFUNC= LEFT		00001880
	LVVARG=LV1 AAE		00001890
	CALL LVLDLET		00001900
	LV1 AAE = LV1 AAD		00001910
C****	LV1 AAE - RIGHT		00001920
	LVVAD=-1		00001930
	LVVTYP=-1		00001940
	LVVPOS=1		00001950
	LVFUNC= RIGHT		00001960
	LVVARG=LV1 AAE		00001970
	CALL LVLDLET		00001980
	LV1 AAE = LV1 AAD		00001990
C****	LV1 AAE - HOL		00002000
	LVVAD=-1		00002010
	LVVTYP=-1		00002020
	LVVPOS=1		00002030
	LVFUNC= HOL		00002040
	LVVARG=LV1 AAE		00002050
	CALL LVLDLET		00002060
	LV1 AAE = LV1 AAD		00002070
C****	LV1 AAE - STRING		00002080
	LVVAD=-1		00002090
	LVVTYP=-1		00002100
	LVVPOS=1		00002110
	LVFUNC= STRING		00002120
	LVVARG=LV1 AAE		00002130
	CALL LVLDLET		00002140
	LVVTR = LVVAL		00002150
	LVVAL = -100		00002160
	IF (LVVTR,EQ,-1) GO TO 30		00002170
	IF (LVVTR,NE,-1) GO TO 30		00002180
35	CONTINUE		00002190
C	STRING=STRING		00002200
C****	STRING - STRING		00002210



LVVAD=-1		00002220
LVVTYP=-1		00002230
LVVPOS=1		00002240
LVFUNC= STRING		00002250
LVVARG= STRING		00002260
CALL LVDLET		00002270
C OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)		00002280
LV1 AAD = OPRAND		00002290
C**** LV1 AAD - OPRAND		00002300
LVVAD=-1		00002310
LVVTYP=-1		00002320
LVVPOS=1		00002330
LVFUNC= OPRAND		00002340
LVVARG=LV1 AAD		00002350
CALL LVDLET		00002360
C**** LV1 AAD - STRING		00002370
LVVAD=-1		00002380
LVVTYP=-1		00002390
LVVPOS=1		00002400
LVFUNC= STRING		00002410
LVVARG=LV1 AAD		00002420
CALL LVDLET		00002430
C**** LV1 AAD - ACTION		00002440
LVVAD=-1		00002450
LVVTYP=-1		00002460
LVVPOS=1		00002470
LVFUNC= ACTION		00002480
LVVARG=LV1 AAD		00002490
CALL LVDLET		00002500
C**** LV1 AAD - FUNC1		00002510
LVVAD=-1		00002520
LVVTYP=-1		00002530
LVVPOS=1		00002540
LVFUNC= FUNC1		00002550
LVVARG=LV1 AAD		00002560
CALL LVDLET		00002570
LVVAD=-1		00002580
LVVTYP=-1		00002590
LVVPOS=1		00002600
LVFUNC= FUNC2		00002610
LVVARG=LV1 AAD		00002620
CALL LVDLET		00002630
LVVAD=-1		00002640
LVVTYP=-1		00002650
LVVPOS=1		00002660
LVFUNC= FUNC3		00002670
LVVARG=LV1 AAD		00002680
CALL LVDLET		00002690
LVVAD=-1		00002700
LVVTYP=-1		00002710
LVVPOS=1		00002720
LVFUNC= LEVEL		00002730
LVVARG=LV1 AAD		00002740
CALL LVDLET		00002750
NSTR=NCHRP		00002760



RETURN	00002770
40 PRINT 300,MAXJ	00002780
300 FORMAT(1X,34H PARSE FAILED AFTER CHARACTER NO. ,I3)	00002790
ERRFLG=.TRUE.	00002800
GO TO 25	00002810
C COMPLETE	00002820
25000 CONTINUE	00002830
LV2A=0	00002840
LV2B=0	00002850
LV2C=0	00002860
LV2D=0	00002870
LV2E=0	00002880
LV2F=0	00002890
LV2G=0	00002900
LV2H=0	00002910
GO TO 25001	00002920
END	00002930

SUBROUTINE PHONEY	00000010
INTEGER FLGSPC,FLAGSP	00000020
COMMON/LVVTR1/LVVSZE,LVVGSP,NODSPC(1000)/LVVTR2/LSTSPC(1000)	00000030
COMMON/LVVTR3/LNKSPC(1000)/LVVTR4/FLGSPC(1000)	00000040
COMMON/LVVTR5/LVFILE,LVCMPR,NODESP( 1)	00000050
*/LVVTR6/LISTSP( 1)/LVVTR7/LINKSP( 1)	00000060
*/LVVTR8/FLAGSP( 1)	00000070
COMMON/LVRAND/LVRPRM,LVKS,LVKX,LVKDY,LVKDX,LVTEMP	00000080
COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPUS,LVVTYP,LVVAL,	00000090
*LVHEAD,LVVNL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP	00000100
COMMON/LVTAHL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSI7E,LVSQSP( 1)	00000110
C EXECUTE	00000120
LVVSZE=1000	00000130
LVFILE=0	00000140
LVCMPR= 0	00000150
LVSI7E= 1	00000160
LVSKIP=1	00000170
LVRPRM=17	00000180
CALL LVSETP	00000190
RETURN	00000200
END	00000210



	SUBROUTINE PPNTS	00000010
	COMMON/LVARG/LVFUNC,LVVAR,LVVAD,LVVPOS,LVVTP,LVVAL,	00000020
	*LVHEAD,LVVNL,LVUEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
	COMMON/LVTAML/LVTSIZ,LVMAP( 1)/LVVSEG/LVSI7E,LVSOSP( 1)	00000040
	COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING	00000050
	COMMON/VAR/VFOR,NCHAR,NCHARP,CHAR,NDICT	00000060
	COMMON/TYP/NAHAY,TYPE1,TYPE2,ERRFLG,LHSTYP	00000070
	COMMON /STRING/ NTYPE,NSTR,STR	00000080
	COMMON /GIML/ NNN(14),OPRAND	00000090
	LOGICAL ERRFLG	00000100
	INTEGER VFOR(30),CHAR,STRING,HOL,RIGHT,STR(1),OPRAND	00000110
C	EXECUTE	00000120
	GO TO 25000	00000130
25001	CONTINUE	00000140
	NCHAR=0	00000150
	NCHARP=0	00000160
	DO 5 I=1,30	00000170
5	VFOR(I)=0	00000180
	NINT=1	00000190
	NTMP=1	00000200
	DO 10 I=1,NSTR	00000210
C	STRING(+HOL,I "NODE,+STRING,I "N1=OPRAND/16)	00000220
	LV1 AAD = STRING	00000230
C****	LV1 AAD + HOL	00000240
	LVPPOS = I	00000250
	LVTYP = 3	00000260
	LVFUNC= HOL	00000270
	LVVAR= LV1 AAD	00000280
	CALL LVFIND(LV2 A, LV2 H, LV2 C, LV2 D)	00000290
	LV1 AAE = LV1 AAD	00000300
	IF (LVVAL.NE.-1) LV1 AAE = LVVAL	00000310
C****	LV1 AAE " NODE	00000320
	NODE = LV1 AAE	00000330
C****	LV1 AAD + STRING	00000340
	LVPPOS = I	00000350
	LVTYP=3	00000360
	LVFUNC= STRING	00000370
	LVVAR= LV1 AAD	00000380
	CALL LVFIND(LV2 E, LV2 F, LV2 G, LV2 H)	00000390
	LV1 AAE = LV1 AAD	00000400
	IF (LVVAL.NE.-1) LV1 AAE = LVVAL	00000410
C****	LV1 AAE " N1	00000420
	N1 = LV1 AAE	00000430
C****	N1 = OPRAND	00000440
	LVVAL = -100	00000450
	IF ( N1.NE. OPRAND) LVVAL = -1	00000460
	LVVTR = LVVAL	00000470
	LVVAL = -100	00000480
	IF (LVVTR.EQ.-1) GO TO 16	00000490
	NINT=NINT+1	00000500
16	J=0	00000510
20	J=J+1	00000520
C	NODE=LEFT,J/30 (+HOL,1 "CHAR,+HOL,2 "NDICT)	00000530
C****	NODE + LEFT	00000540
	LVPPOS = J	00000550



LVVTYP=3		00000560
LVFUNC= LEFT		00000570
LVVARG= NODE		00000580
CALL LVFIND(LV2	I,LV2 J,LV2 K,LV2 L)	00000590
LV1 AAD =	NODE	00000600
IF (LVVAL.NE.-1) LV1	AAI) = LVVAL	00000610
LVVTH = LVVAL		00000620
LVVAL = -100		00000630
IF (LVVTH.EQ.-1) GO TO	30	00000640
LV1 AAE = LV1	AAO	00000650
C**** LV1 AAE	+ HOL	00000660
LVVPOS =	1	00000670
LVVTYP =	3	00000680
LVFUNC=	HOL	00000690
LVVARG= LV1	AAE	00000700
CALL LVFIND(LV2	M,LV2 N,LV2 O,LV2 P)	00000710
LV1 AAF = LV1	AAE	00000720
IF (LVVAL.NE.-1) LV1	AAF = LVVAL	00000730
C**** LV1 AAF	" CHAR	00000740
LV1 CHAR = LV1	AAF	00000750
LV1 AAF = LV1	AAO	00000760
C**** LV1 AAF	+ HOL	00000770
LVVPOS =	2	00000780
LVVTYP =	3	00000790
LVFUNC=	HOL	00000800
LVVARG= LV1	AAF	00000810
CALL LVFIND(LV2	Q,LV2 R,LV2 S,LV2 T)	00000820
LV1 AAE = LV1	AAF	00000830
IF (LVVAL.NE.-1) LV1	AAE = LVVAL	00000840
C**** LV1 AAE	" NDICT	00000850
NDICT = LV1	AAE	00000860
CALL FORM		00000870
GO TO 20		00000880
30 CONTINUE		00000890
IF (NINT .GT. NTMP) GO TO 35		00000900
C N1(+HOL.1 "CHAR,+HOL.2 "NDICT)		00000910
LV1 AAD =	N1	00000920
C**** LV1 AAD	+ HOL	00000930
LVVPOS =	1	00000940
LVVTYP =	3	00000950
LVFUNC=	HOL	00000960
LVVARG= LV1	AAO	00000970
CALL LVFIND(LV2	U,LV2 V,LV2 W,LV2 X)	00000980
LV1 AAE = LV1	AAO	00000990
IF (LVVAL.NE.-1) LV1	AAE = LVVAL	00010000
C**** LV1 AAE	" CHAR	00010010
LV1 CHAR = LV1	AAE	00010020
C**** LV1 AAD	+ HOL	00010030
LVVPOS =	2	00010040
LVVTYP =	3	00010050
LVFUNC=	HOL	00010060
LVVARG= LV1	AAO	00010070
CALL LVFIND(LV2	Y,LV2 Z,LV2 O,LV2 I)	00010080
LV1 AAE = LV1	AAO	00010090
IF (LVVAL.NE.-1) LV1	AAE = LVVAL	00010100



C****	LV1	AAE	"	NDICT	00001110
		NDICT = LV1	AAE		00001120
		CALL FORM			00001130
		GO TO 37			00001140
	35	NTMP=NINT			00001150
C		NODE+HOL "NDICT			00001160
C****		NODE	+	HOL	00001170
		LVVTYP = 3			00001180
		LVVPOS = 1			00001190
		LVINDX = 0			00001200
		LVFUNC=	HOL		00001210
		LVVARG=	NODE		00001220
		CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)			00001230
		LV1	AAE =	NODE	00001240
		IF (LVVAL, NE, -1) LV1	AAE =	LVVAL	00001250
C****	LV1	AAE	"	NDICT	00001260
		NDICT = LV1	AAE		00001270
C		OPRAND+HOL "CHAR			00001280
C****		OPRAND	+	HOL	00001290
		LVVTYP = 3			00001300
		LVVPOS = 1			00001310
		LVINDX=0			00001320
		LVFUNC=	HOL		00001330
		LVVARG=	OPRAND		00001340
		CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)			00001350
		LV1	AAE =	OPRAND	00001360
		IF (LVVAL, NE, -1) LV1	AAE =	LVVAL	00001370
C****	LV1	AAE	"	CHAR	00001380
		CHAR = LV1	AAE		00001390
		CALL FORM			00001400
	37	J=0			00001410
	40	J=J+1			00001420
C		NODE+RIGHT, J/10 (+HOL, 1 "CHAR, +HOL, 2 "NDICT)			00001430
C****		NODE	+	RIGHT	00001440
		LVVPOS =	J		00001450
		LVVTYP = 3			00001460
		LVFUNC=	RIGHT		00001470
		LVVARG=	NODE		00001480
		CALL LVFIND(LV2	2, LV2	3, LV2	4, LV2
		LV1	AAE =	NODE	5)
		IF (LVVAL, NE, -1) LV1	AAE =	LVVAL	
		LVVTR = LVVAL			
		LVVAL = -100			
		IF (LVVTR, EQ, -1) GO TO		10	
		LV1	AAE = LV1	AAE	
C****	LV1	AAE	+	HOL	
		LVVPOS =	1		
		LVVTYP=3			
		LVFUNC=	HOL		
		LVVARG= LV1	AAE		
		CALL LVFIND(LV2	6, LV2	7, LV2	8, LV2
		LV1	AAE = LV1	AAE	9)
		IF (LVVAL, NE, -1) LV1	AAE =	LVVAL	
C****	LV1	AAE	"	CHAR	
		CHAR = LV1	AAE		



C****	LV1	AAF = LV1	AAD			00001660
	LV1	AAF	*	HOL		00001670
	LVVPOS =	2				00001680
	LVVTYP =	3				00001690
	LVFUNC=	HOL				00001700
	LVVARG= LV1	AAF				00001710
	CALL LVFIND(LV2	AA, LV2	AB, LV2	AC, LV2	AD)	00001720
	LV1	AAE = LV1	AAF			00001730
	IF (LVVAL.NE.-1)	LV1	AAE = LVVAL			00001740
C****	LV1	AAE	"	NDICT		00001750
	NDICT = LV1	AAE				00001760
	CALL FORM					00001770
	GO TO 40					00001780
10	CONTINUE					00001790
	NC=1+(NCHARP-1)/4					00001800
100	FORMAT(1X,30A4)					00001810
	IF (ERRFLG) PRINT 100, (VFOR(I), I=1, NC)					00001820
C	COMPLETE					00001830
	RETURN					00001840
25000	CONTINUE					00001850
	LV2A=0					00001860
	LV2B=0					00001870
	LV2C=0					00001880
	LV2D=0					00001890
	LV2E=0					00001900
	LV2F=0					00001910
	LV2G=0					00001920
	LV2H=0					00001930
	LV2I=0					00001940
	LV2J=0					00001950
	LV2K=0					00001960
	LV2L=0					00001970
	LV2M=0					00001980
	LV2N=0					00001990
	LV2O=0					00002000
	LV2P=0					00002010
	LV2Q=0					00002020
	LV2R=0					00002030
	LV2S=0					00002040
	LV2T=0					00002050
	LV2U=0					00002060
	LV2V=0					00002070
	LV2W=0					00002080
	LV2X=0					00002090
	LV2Y=0					00002100
	LV2Z=0					00002110
	LV20=0					00002120
	LV21=0					00002130
	LV22=0					00002140
	LV23=0					00002150
	LV24=0					00002160
	LV25=0					00002170
	LV26=0					00002180
	LV27=0					00002190
	LV28=0					00002200
	LV29=0					00002210
	LV2AA=0					00002220
	LV2AB=0					00002230
	LV2AC=0					00002240
	LV2AD=0					00002250
	GO TO 25001					00002260
	END					00002270



SUBROUTINE PROG	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
IDTYP=2	00000050
CALL STORE	00000060
RETURN	00000070
END	00000080



### 31 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFFFE/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q10PRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFFE/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q10PRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFFE/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

### 30 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFFC/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q10PRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFC/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q10PRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFFC/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```



## 29 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF8/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF8/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF8/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

## 28 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF0/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```



27 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFE0/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFE0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFE0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

26 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFC0/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFC0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFC0/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```



## 25 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF80/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF80/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF80/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

## 24 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/ZFFFFFFF00/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF00/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/ZFFFFFFF00/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```



SUBROUTINE REALCK	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCM(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IOTYP,NID,	00000030
• LOC,LJYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/LOGIC/LOG,LOGST	00000050
COMMON/REALNO/IREAL,IRELND,IP	00000060
INTEGER A,DECPT,EEE,PLUS,MINUS,DEE,ELOC	00000070
DATA DECPT/1H./,EEE/1HE/,PLUS/1H./,MINUS/1H-/,DEE/1HD/	00000080
IDES=0	00000090
IRELND=0	00000100
IF(IP .GE. N) GO TO 90	00000110
JPTR=IP	00000120
IF(NEXT(JPTR) .EQ. DECPT) GO TO 5	00000130
JPTR=IP	00000140
IF(ITYPE(JPTR) .EQ. 2) GO TO 10	00000150
GO TO 90	00000160
5 IF(JPTR .GT. N) GO TO 90	00000170
IF(ITYPE(JPTR) .NE. 2) GO TO 90	00000180
GO TO 20	00000190
10 IF(JPTR .GT. N) GO TO 90	00000200
12 IF(ITYPE(JPTR) .EQ. 2) GO TO 15	00000210
IF(A(JPTR-1) .NE. DECPT) GO TO 90	00000220
LOGST=JPTR	00000230
CALL LOGCHK	00000240
IF(LOG .EQ. 1) GO TO 90	00000250
JPTR=LOGST	00000260
GO TO 20	00000270
15 IF(JPTR .GT. N) GO TO 90	00000280
GO TO 12	00000290
20 IREAL=1	00000300
IF(JPTR .GT. N) GO TO 35	00000310
22 IF(ITYPE(JPTR) .EQ. 2) GO TO 25	00000320
IF(A(JPTR-1) .EQ. EEE) GO TO 24	00000330
IF(A(JPTR-1) .NE. DEE) GO TO 30	00000340
IDES=1	00000350
24 ELOC=JPTR-2	00000360
GO TO 40	00000370
25 IF(JPTR .GT. N) GO TO 35	00000380
GO TO 22	00000390
30 ELOC=JPTR-2	00000400
32 IRELND=ELOC	00000410
RETURN	00000420
35 IRELND=N	00000430
RETURN	00000440
40 IF(JPTR .GT. N) GO TO 32	00000450
NXT=NEXT(JPTR)	00000460
IF(NXT .EQ. PLUS .OR. NXT .EQ. MINUS) GO TO 45	00000470
JPTR=JPTR-1	00000480
IF(ITYPE(JPTR) .NE. 2) GO TO 32	00000490
IF(JPTR .GT. N) GO TO 35	00000500
GO TO 47	00000510
45 IF(JPTR .GT. N) GO TO 32	00000520
IF(ITYPE(JPTR) .NE. 2) GO TO 32	00000530
47 IF(ITYPE(JPTR) .EQ. 2) GO TO 50	00000540
IRELND=JPTR-2	00000550
RETURN	00000560
50 IF(JPTR .GT. N) GO TO 35	00000570
GO TO 47	00000580
90 IREAL=0	00000590
RETURN	00000600
END	00000610



```

SUBROUTINE RECOG(FIN)                                00000010
COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTP,LVVAL, 00000020
*LVHEAD,LVVNL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP    00000030
COMMON/LVTABL/LVTSIZ,LVMAP(1)/LVVSEQ/LVSIZE,LVSEQSP  00000040
COMMON/HL/HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ 00000050
COMMON/NEED/START,ASSOC,LEVEL,STOP                   00000060
COMMON/STRING/NNN(2),STR                             00000070
COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4) 00000080
INTEGER HOL                                           00000150
INTEGER START,ASSOC,STOP,RETRN,R,STJ,STACK,STR(1),ACTION,STRING, 00000160
* RTEMP                                               00000170
LOGICAL FAIL,FIN                                     00000180
FIN=.FALSE.                                          00000190
C EXECUTE                                           00000200
GO TO 25000                                         00000210
25001 CONTINUE                                       00000220
JSTACK=0                                           00000230
J=1                                                00000240
C START "R                                           00000250
C****      START      "      R                    00000260
          R =      START                                00000270
C STRING+STRING,J/70 "STJ                          00000280
C****      STRING      +      STRING                00000290
          LVVPOS =      J                                00000300
          LVVTP =      3                                00000310
          LVFUNC=      STRING                          00000320
          LVVARG=      STRING                          00000330
          CALL LVFIND(LV2      A,LV2      B,LV2      C,LV2      D) 00000340
          LV1 AAD =      STRING                          00000350
          IF (LVVAL,NE,-1) LV1 AAD = LVVAL              00000360
          LVVTR = LVVAL                                00000370
          LVVAL = -100                                00000380
          IF (LVVTR,EQ,-1) GO TO      70                00000390
C**** LV1 AAD "      STJ                            00000400
          STJ = LV1 AAD                                00000410
          M=-1                                         00000420
C STRING+HOL,1 STRING ""M""                          00000430
C****      STRING      +      HOL                    00000440
          LVVPOS =      1                                00000450
          LVVTP=3                                       00000460
          LVFUNC=      HOL                              00000470
          LVVARG=      STRING                          00000480
          CALL LVFIND(LV2      E,LV2      F,LV2      G,LV2      H) 00000490
          LV1 AAD =      STRING                          00000500
          IF (LVVAL,NE,-1) LV1 AAD = LVVAL              00000510
C**** LV1 AAD STRING ""                                     00000520
          LVDEST= 0                                    00000530
          LV1 AAE = M                                    00000540
          LVTYPE(1) = 1                                00000550
          LVVALS(1) = LV1 AAE                          00000560
          LVDEST= 0                                    00000570
          LVVNL = 1                                    00000580
          LVFUNC =      STRING                          00000590
          LVVARG=LV1 AAD                                00000600
          CALL LVNSRT                                  00000610

```



IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00000620
IF (LVVAL.LT.0) RETURN	00000630
6 CONTINUE	00000640
C 10 R(+ASSOC//15,+STOP//15,=STOP/20)	00000650
10 CONTINUE	00000660
LV1 AAD = R	00000670
C**** LV1 AAD + ASSOC	00000680
LVVTYP = 3	00000690
LVVPOS = 1	00000700
LVINDX = 0	00000710
LVFUNC= ASSOC	00000720
LVVARG= LV1 AAD	00000730
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00000740
LV1 AAF = LV1 AAD	00000750
IF (LVVAL.NE.-1) LV1 AAF = LVVAL	00000760
LVVTR = LVVAL	00000770
LVVAL = -100	00000780
IF (LVVTR.NE.-1) GO TO 15	00000790
C**** LV1 AAD + STOP	00000800
LVVTYP = 3	00000810
LVVPOS = 1	00000820
LVINDX = 0	00000830
LVFUNC= STOP	00000840
LVVARG= LV1 AAD	00000850
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00000860
LV1 AAF = LV1 AAD	00000870
IF (LVVAL.NE.-1) LV1 AAF = LVVAL	00000880
LVVTR = LVVAL	00000890
LVVAL = -100	00000900
IF (LVVTR.NE.-1) GO TO 15	00000910
LVVAL = -100	00000920
IF (LV1 AAD.NE. STOP) LVVAL = -1	00000930
LVVTR = LVVAL	00000940
LVVAL = -100	00000950
IF (LVVTR.EQ.-1) GO TO 20	00000960
15 JSTACK=JSTACK+1	00000970
STACK(JSTACK,1)=R	00000980
STACK(JSTACK,2)=0	00000990
STACK(JSTACK,3)=J	00001000
STACK(JSTACK,4)=0	00001010
C 20 R+ACTION/22 "N	00001020
20 CONTINUE	00001030
C**** R + ACTION	00001040
LVVTYP = 3	00001050
LVVPOS = 1	00001060
LVINDX = 0	00001070
LVFUNC= ACTION	00001080
LVVARG= R	00001090
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00001100
LV1 AAD = R	00001110
IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00001120
LVVTR = LVVAL	00001130
LVVAL = -100	00001140
IF (LVVTR.EQ.-1) GO TO 22	00001150
C**** LV1 AAD " N	00001160



N = LV1	AAD	00001170
CALL SEMANT(N,FAIL)		00001180
IF (FAIL) GO TO 99		00001190
GO TO 25		00001200
C 22 R+STJ/99 "R		00001210
22 CONTINUE		00001220
C**** R	STJ	00001230
LVVTYP = 3		00001240
LVVPOS = 1		00001250
LVINDX = 0		00001260
LVFUNC = STJ		00001270
LVVARG = R		00001280
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)		00001290
LV1 AAD = R		00001300
IF (LVVAL, NE, -1) LV1 AAD = LVVAL		00001310
LVVTR = LVVAL		00001320
LVVAL = -100		00001330
IF (LVVTR, EQ, -1) GO TO 99		00001340
C**** LV1 AAD "	R	00001350
R = LV1 AAD		00001360
25 J=J+1		00001370
IF (J, GT, MAXJ) MAXJ=J		00001380
C 30 STRING+STRING, J "STJ//6		00001390
30 CONTINUE		00001400
C**** STRING	STRING	00001410
LVVPOS = J		00001420
LVVTYP = 3		00001430
LVFUNC = STRING		00001440
LVVARG = STRING		00001450
CALL LVFIND(LV2 I, LV2 J, LV2 K, LV2 L)		00001460
LV1 AAD = STRING		00001470
IF (LVVAL, NE, -1) LV1 AAD = LVVAL		00001480
C**** LV1 AAD "	STJ	00001490
STJ = LV1 AAD		00001500
LVVTR = LVVAL		00001510
LVVAL = -100		00001520
IF (LVVTR, NE, -1) GO TO 6		00001530
40 STJ=-1		00001540
C R+ACTION/42 "N		00001550
C**** R	ACTION	00001560
LVVTYP = 3		00001570
LVVPOS = 1		00001580
LVINDX = 0		00001590
LVFUNC = ACTION		00001600
LVVARG = R		00001610
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)		00001620
LV1 AAD = R		00001630
IF (LVVAL, NE, -1) LV1 AAD = LVVAL		00001640
LVVTR = LVVAL		00001650
LVVAL = -100		00001660
IF (LVVTR, EQ, -1) GO TO 42		00001670
C**** LV1 AAD "	N	00001680
N = LV1 AAD		00001690
CALL SEMANT(N,FAIL)		00001700
IF (FAIL) GO TO 99		00001710



42 CALL SSTOP(FAIL)	00001720
IF(FAIL) GO TO 99	00001730
JSTACK=JSTACK+1	00001740
STACK(JSTACK,1)=R	00001750
STACK(JSTACK,2)=0	00001760
STACK(JSTACK,3)=J	00001770
STACK(JSTACK,4)=0	00001780
C R=ACTION/44 "N	00001790
C**** R * ACTION	00001800
LVVTYP = 3	00001810
LVVPOS = 1	00001820
LVINDX = 0	00001830
LVFUNC= ACTION	00001840
LVVARG= R	00001850
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00001860
LV1 AAD = R	00001870
IF (LVVAL,NE,-1) LV1 AAD = LVVAL	00001880
LVVTR = LVVAL	00001890
LVVAL = -100	00001900
IF (LVVTR,EQ,-1) GO TO 44	00001910
C**** LV1 AAD " N	00001920
N = LV1 AAD	00001930
CALL SEMANT(N,FAIL)	00001940
IF(FAIL) GO TO 99	00001950
44 CALL SLEVEL(FAIL)	00001960
IF(FAIL) RETURN	00001970
GO TO 40	00001980
99 CONTINUE	00001990
CALL RECOV(RETRN)	00002000
IF(RETRN,LT, 0) GO TO 70	00002010
C RETRN=ASSOC/30	00002020
C**** RETRN = ASSOC	00002030
LVVAL = -100	00002040
IF ( RETRN,NE, ASSOC) LVVAL = -1	00002050
LVVTR = LVVAL	00002060
LVVAL = -100	00002070
IF (LVVTR,NE,-1) GO TO 30	00002080
IF(RETRN,EQ, 0) GO TO 10	00002090
CALL SLEVEL(FAIL)	00002100
IF(FAIL) GO TO 65	00002110
GO TO 30	00002120
65 IF(JSTACK,LE, 1) GO TO 70	00002130
JSTACK=JSTACK-1	00002140
GO TO 99	00002150
70 FIN=.TRUE.	00002160
RETURN	00002170
C COMPLETE	00002180
25000 CONTINUE	00002190
LV2A=0	00002200
LV2B=0	00002210
LV2C=0	00002220
LV2D=0	00002230
LV2E=0	00002240
LV2F=0	00002250
LV2G=0	00002260
LV2H=0	00002270
LV2I=0	00002280
LV2J=0	00002290
LV2K=0	00002300
LV2L=0	00002310
GO TO 25001	00002320
END	00002330



SUBROUTINE RECOV(RETRN)	00000010
COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTP,LVVAL,	00000020
*LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
COMMON/LVTABL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSGSP( 1)	00000040
COMMON /NEED/ START,ASSOC,LEVEL,STOP	00000050
COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)	00000060
COMMON /STRING/ NNN(2),STR	00000070
COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING	00000080
INTEGER START,ASSOC,STOP,STACK,STR(1),STJ,R,STRING,RETRN,TEMP	00000090
*,RIGHT,HOL	00000100
C EXECUTE	00000110
GO TO 25000	00000120
25001 CONTINUE	00000130
10 R=STACK(JSTACK,1)	00000140
JAS=STACK(JSTACK,2)+1	00000150
C R=ASSOC,JAS "TEMP//30	00000160
C**** R * ASSOC	00000170
LVVPOS = JAS	00000180
LVVTP = 3	00000190
LVFUNC= ASSOC	00000200
LVVARG= R	00000210
CALL LVFIND(LV2 A,LV2 B,LV2 C,LV2 D)	00000220
LV1 AAD = R	00000230
IF (LVVAL,NE,-1) LV1 AAD = LVVAL	00000240
C**** LV1 AAD " TEMP	00000250
TEMP = LV1 AAD	00000260
LVVTR = LVVAL	00000270
LVVAL = -100	00000280
IF (LVVTR,NE,-1) GO TO 30	00000290
C 15 R=(STOP//40)+STOP/16=STOP//40 "R)	00000300
15 CONTINUE	00000310
LV1 AAD = R	00000320
C**** LV1 AAD = STOP	00000330
LVVAL = -100	00000340
IF (LV1 AAD,NE, STOP) LVVAL = -1	00000350
LVVTR = LVVAL	00000360
LVVAL = -100	00000370
IF (LVVTR,NE,-1) GO TO 40	00000380
C**** LV1 AAD * STOP	00000390
LVVTP = 3	00000400
LVVPOS = 1	00000410
LVINDX = 0	00000420
LVFUNC= STOP	00000430
LVVARG= LV1 AAD	00000440
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00000450
LV1 AAF = LV1 AAD	00000460
IF (LVVAL,NE,-1) LV1 AAF = LVVAL	00000470
LVVTR = LVVAL	00000480
LVVAL = -100	00000490
IF (LVVTR,EQ,-1) GO TO 16	00000500
C**** LV1 AAF = STOP	00000510
LVVAL = -100	00000520
IF (LV1 AAF,NE, STOP) LVVAL = -1	00000530
LVVTR = LVVAL	00000540
LVVAL = -100	00000550



IF (LVVTR,NE,-1) GO TO	40	00000560
C**** LV1    AAF    "    R		00000570
R = LV1    AAF		00000580
J=STACK(JSTACK,3)		00000590
JSTACK=JSTACK-1		00000600
RETRN=0		00000610
RETURN		00000620
16 JSTACK=JSTACK-1		00000630
IF(JSTACK,LE,0) GO TO 20		00000640
IF(STACK(JSTACK,3),LT,J) CALL SEMANT(0,FAIL)		00000650
J=STACK(JSTACK,3)		00000660
GO TO 10		00000670
20 RETRN=-1		00000680
RETURN		00000690
40 CONTINUE		00000700
J=STACK(JSTACK,3)		00000710
IF(STACK(JSTACK,4),GT,0) GO TO 16		00000720
RETRN=STOP		00000730
RETURN		00000740
C 30 TEMP "R		00000750
30 CONTINUE		00000760
C****    TEMP    "    R		00000770
R =    TEMP		00000780
IF(JSTACK,EQ,1) GO TO 35		00000790
IF( R,NE,STACK(JSTACK,1)) GO TO 35		00000800
NTEMP=STACK(JSTACK-1,1)		00000810
JMARK=JSTACK		00000820
31 STACK(JMARK,4)=-1		00000830
JMARK=JMARK-1		00000840
IF(R,EQ,STACK(JMARK,1),AND,STACK(JMARK,4),LT,0) GO TO 15		00000850
IF(R,EQ,STACK(JMARK,1),AND,JMARK,NE,0) GO TO 31		00000860
IF(R,NE,NTEMP,OR,JAS,NE,STACK(JSTACK-1,2),OR,		00000870
* STACK(JSTACK-1,4),GE,0) GO TO 35		00000880
GO TO 15		00000890
35 CONTINUE		00000900
IF(STACK(JSTACK,3),LT,J) CALL SEMANT(0,FAIL)		00000910
STACK(JSTACK,2)=JAS		00000920
J=STACK(JSTACK,3)		00000930
IF(STACK(JSTACK,4),GT,0) STACK(JSTACK,4)=0		00000940
RETRN=ASSOC		00000950
RETURN		00000960
C COMPLETE		00000970
25000 CONTINUE		00000980
LV2A=0		00000990
LV2F=0		00001000
LV2C=0		00001010
LV2D=0		00001020
GO TO 25001		00001030
END		00001040



```

SUBROUTINE ROLCHK(I1,I2,I3,I4,I5,I6)
DIMENSION ARG(6),SUBNM(2),ISUBNM(2)
EQUIVALENCE (ISUBNM(1),SUBNM(1))
DATA MASK/ZFF00'000/
ARG(1)=AND(I1,MASK)
ARG(2)=AND(I2,MASK)
ARG(3)=AND(I3,MASK)
ARG(4)=AND(I4,MASK)
ARG(5)=AND(I5,MASK)
ARG(6)=AND(I6,MASK)
ISUBNM(1)=0
ISUBNM(2)=0
NCHAR=0
J=0
10 J=J+1
DO 30 I=1,4
NCHAR=NCHAR+1
IF(I.EQ. 1) GO TO 20
NSHIFT=8*(I-1)
CALL SHIFTR(ARG(NCHAR),NSHIFT)
20 SUBNM(J)=OR(SUBNM(J),ARG(NCHAR))
IF(NCHAR.EQ. 6) GO TO 35
30 CONTINUE
GO TO 10
35 WRITE(3,40) ISUBNM(1),ISUBNM(2)
40 FORMAT(A4,A2)
RETURN
END

```

```

SUBROUTINE SEARCH
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000010
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000020
* LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES 00000030
DO 20 IDTYP=1,2 00000040
J=INITID(IDTYP) 00000050
IF(J.EQ. 0) GO TO 15 00000060
DO 10 I=1,NID 00000070
IF(ICOMP(NXTID,IDTBL,J,11).EQ. 0) GO TO 5 00000080
ISRCH(IDTYP)=1 00000090
IDES=LOC 00000100
LOC=J 00000110
GO TO 20 00000120
5 J=IDTBL(4,J) 00000130
IF(J.EQ. 0) GO TO 15 00000140
10 CONTINUE 00000150
15 ISRCH(IDTYP)=0 00000160
20 CONTINUE 00000170
RETURN 00000180
END 00000190
00000200

```



```

SUBROUTINE SEMANT(N,FAIL)
COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYPE,LVVAL
*LVH*AD,LVVNVL,LVUEST,LVVALS(10),LVTYPE(10),LVSKIP
COMMON/LVTABL/LVTSIZ,LVMAP(1)/LVVSEQ/LVSIZE,LVSGSP(1)
COMMON/FUNC/ NARY(5,22),MARGS,IARGS(50),FNCLC(5),NFUNC
COMMON/HL/HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
COMMON /TYP/ NARWAY,TYPE1,TYPE2,ERRFLG
COMMON /STRING/ NTYPE,NSTR,STR
COMMON /JL/ JSTOP
COMMON /GIRL/NTERMS,PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,
*LE,GT,GE,EQ,NE,OR,AND,NOT,EQUALS,OPRAND
COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
COMMON /NEED/ START,ASSOC,LEVEL,STOP
COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG
INTEGER HOL,ACTION,FUNC,LEFT,RIGHT,STRING,RPAR,STJ,R,STACK
$ ,EXP,FUNC1,FUNC2,FUNC3,TYPE1,TYPE2,TYPE(2,5),STR(1),STOP
$ ,ALPHA,BETA,GAMMA,OPRAND,EQUALS,AND,OR,COMMA
LOGICAL SKIP,FLAG,ERRFLG,FAIL,NOTFLG
INTEGER FUNCRF,BITPUT,PLUS,FL(3),BITGET
INTEGER GETTYP,GETDIM
DATA FLAG/,FALSE/,FUNCRF/86/
DATA TYPE/4HREAL,1H ,4HCOMP,3HLEX,4HDOUH,
* 2HLE,4HINTE,3HGER,4HLOGI,3HCAL/
GETTYP(II)=MOD(II,100000)/10000
GETDIM(II)=MOD(II,1000000)/100000
C EXECUTE
GO TO 25000
25001 CONTINUE
FAIL=.FALSE.
IF(N, EQ, 0) GO TO 999
GO TO(10,20,30,40,50,60,70,80,90,1000,1100,1200,1300,1400,1500,
$ 1600),N
C 10 R=STJ/11 "R//12
10 CONTINUE
C**** R * STJ
LVVTYP = 3
LVVPOS = 1
LVINDX = 0
LVFUNC= STJ
LVVARG= P
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
LV1 AAD = R
IF (LVVAL,NE,-1) LV1 AAD = LVVAL
LVVTR = LVVAL
LVVAL = -100
IF (LVVTR,EQ,-1) GO TO 11
C**** LV1 AAD " R
H = LV1 AAD
LVVTR = LVVAL
LVVAL = -100
IF (LVVTR,NE,-1) GO TO 12
11 FAIL=.TRUE.
RETURN
C PRIMARY RECOGNIZED
12 IF(STJ,EQ, PLUS ,OR, STJ,EQ, MINUS) GO TO 126

```



IF(STJ,NE,PPAR) GO TO 121	00000560
JSTACK=JSTACK+1	00000570
STACK(JSTACK,1)=STOP	00000580
STACK(JSTACK,2)=0	00000590
STACK(JSTACK,3)=J	00000600
STACK(JSTACK,4)=0	00000610
NTMP=R	00000620
CALL SLEVEL(SKIP)	00000630
JSTACK=JSTACK-1	00000640
R=NTMP	00000650
JLAST=1	00000660
IF(JSTOP,GT,0) JLAST=STACK(JSTOP,3)	00000670
C STRING+HOL,JLAST(-STRING,STRING "TYPE1")	00000680
C**** STRING * HOL	00000690
LVVPOS = JLAST	00000700
LVVTYP = 3	00000710
LVFUNC= HOL	00000720
LVVARG= STRING	00000730
CALL LVFIND(LV2 A,LV2 H,LV2 C,LV2 D)	00000740
LV1 AAD = STRING	00000750
IF (LVVAL,NE,-1) LV1 AAD = LVVAL	00000760
LV1 AAF = LV1 AAD	00000770
C**** LV1 AAF - STRING	00000780
LVVAD=-1	00000790
LVVTYP=-1	00000800
LVVPOS=1	00000810
LVFUNC= STRING	00000820
LVVARG=LV1 AAF	00000830
CALL LVDELET	00000840
LV1 AAF = LV1 AAD	00000850
C**** LV1 AAF STRING ""	00000860
LVDEST= 0	00000870
LV1 AAG = TYPE1	00000880
LVTYPE(1) = 1	00000890
LVVALS(1) = LV1 AAG	00000900
LVDEST= 0	00000910
LVVNVL = 1	00000920
LVFUNC = STRING	00000930
LVVARG=LV1 AAF	00000940
CALL LVNSRT	00000950
IF(LVVAL,LT,0) CALL LVEXIT(LVVAL)	00000960
IF(LVVAL,LT,0) RETURN	00000970
RETURN	00000980
121 CONTINUE	00000990
C GET TYPE	00001000
BETA=GETDIM(STR(J))	00001010
IF(BETA,NE,5) GO TO 125	00001020
C OPERAND IS A FUNCTION REFERENCE	00001030
IF(NDEP,EQ,0) GO TO 1R	00001040
LVVPOS=-LVVPOS	00001050
LVVTYP= 3	00001060
LVVPOS= 1	00001070
LVDEST= 2	00001080
LV1 AAD = 1	00001090
LVTYPE(1) = 1	00001100



LVVALS(1) = LV1	AAD	00001110
LVDEST = 2		00001120
LVVNVL = 1		00001130
LVFUNC =	FUNC1	00001140
LVVARG =	OPRAND	00001150
CALL LVNSRT		00001160
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)		00001170
IF(LVVAL.LT.0) RETURN		00001180
18 R=FUNCRF		00001190
JSTACK=JSTACK+1		00001200
STACK(JSTACK,1)=R		00001210
STACK(JSTACK,2)=0		00001220
STACK(JSTACK,3)=J+1		00001230
STACK(JSTACK,4)=0		00001240
125 ALPHA=GETTYP(STR(J))		00001250
IF(TYPE1 .GE. 0) GO TO 13		00001260
C SET TYPE OF STATEMENT		00001270
TYPE1=ALPHA		00001280
IF(NTYPE .EQ. 3) TYPE1=-1		00001290
C 126 STRING=HOL.J(-STRING,STRING ""TYPE1"")		00001300
126 CONTINUE		00001310
C****	STRING * HOL	00001320
LVVPOS =	J	00001330
LVVTYP =	3	00001340
LVFUNC=	HOL	00001350
LVVARG=	STRING	00001360
CALL LVFIND(LV2	E,LV2 F,LV2 G,LV2 H)	00001370
LV1 AAD =	STRING	00001380
IF (LVVAL.NE.-1) LV1 AAD = LVVAL		00001390
LV1 AAF = LV1 AAD		00001400
C****	LV1 AAF - STRING	00001410
LVVAD=-1		00001420
LVVTYP=-1		00001430
LVVPOS=1		00001440
LVFUNC=	STRING	00001450
LVVARG=LV1 AAF		00001460
CALL LVULET		00001470
LV1 AAF = LV1 AAD		00001480
C****	LV1 AAF STRING ""	00001490
LVDEST= 0		00001500
LV1 AAM = TYPE1		00001510
LVTYPE(1) = 1		00001520
LVVALS(1) = LV1 AAM		00001530
LVDEST= 0		00001540
LVVNVL = 1		00001550
LVFUNC =	STRING	00001560
LVVARG=LV1 AAF		00001570
CALL LVNSRT		00001580
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)		00001590
IF(LVVAL.LT.0) RETURN		00001600
RETURN		00001610
13 IF(FLAG) GO TO 15		00001620
C CHECK FOR MIXED MODE EXPRESSION		00001630
IF(TYPE1 .EQ. ALPHA .OR. ALPHA .EQ. 5) GO TO 16		00001640
14 NI=TYPE1+1		00001650



N2=ALPHA+1	00001660
ERRFLG=.TRUE.	00001670
CALL ERROR(77,TYPE(1,N1),TYPE(2,N1),TYPE(1,N2),TYPE(2,N2))	00001680
C STRING+HOL,J(-STRING,STRING ""TYPE1"")	00001690
C**** STRING * HOL	00001700
LVVPOS = J	00001710
LVVTYP = 3	00001720
LVFUNC= HOL	00001730
LVVARG= STRING	00001740
CALL LVFIND(LV2 I, LV2 J, LV2 K, LV2 L)	00001750
LV1 AAD = STRING	00001760
IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00001770
LV1 AAF = LV1 AAD	00001780
C**** LV1 AAF - STRING	00001790
LVVAL=-1	00001800
LVVTYP=-1	00001810
LVVPOS=1	00001820
LVFUNC= STRING	00001830
LVVARG=LV1 AAF	00001840
CALL LVULET	00001850
LV1 AAF = LV1 AAD	00001860
C**** LV1 AAF STRING ""	00001870
LVDEST= 0	00001880
LV1 AAI = TYPE1	00001890
LVTYPE(1) = 1	00001900
LVVALS(1) = LV1 AAI	00001910
LVDEST= 0	00001920
LVVNVL = 1	00001930
LVFUNC = STRING	00001940
LVVARG=LV1 AAF	00001950
CALL LVNSRT	00001960
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00001970
IF (LVVAL.LT.0) RETURN	00001980
RETURN	00001990
C PARSING AN EXPONENT	00002000
15 IF ((ALPHA .EQ. 3 .AND. TYPE1 .EQ. 3).OR.((TYPE1 .EQ. 0 .OR. TYPE1	00002010
* .EQ. 2) .AND.	00002020
* (ALPHA .EQ. 0 .OR. ALPHA .EQ. 2))) GO TO 16	00002030
CALL ERROR(78,J,IDM2,IDM3,IDM4)	00002040
ERRFLG=.TRUE.	00002050
C STRING+HOL,J(-STRING,STRING ""TYPE1"")	00002060
C**** STRING * HOL	00002070
LVVPOS = J	00002080
LVVTYP = 3	00002090
LVFUNC= HOL	00002100
LVVARG= STRING	00002110
CALL LVFIND(LV2 M, LV2 N, LV2 O, LV2 P)	00002120
LV1 AAD = STRING	00002130
IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00002140
LV1 AAF = LV1 AAD	00002150
C**** LV1 AAF - STRING	00002160
LVVAD=-1	00002170
LVVTYP=-1	00002180
LVVPOS=1	00002190
LVFUNC= STRING	00002200



```

LVVARG=LV1    AAF                                00002210
CALL LVDLET                                00002220
LV1    AAF = LV1    AAD                                00002230
C**** LV1    AAF    STRING    ""                                00002240
LVDEST= 0                                00002250
LV1    AAJ = TYPE1                                00002260
LVTYPE(1) = 1                                00002270
LVVALS(1) = LV1    AAF                                00002280
LVDEST= 0                                00002290
LVVNVL = 1                                00002300
LVFUNC =    STRING                                00002310
LVVARG=LV1    AAF                                00002320
CALL LVNSRT                                00002330
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)            00002340
IF (LVVAL.LT.0) RETURN                        00002350
RETURN                                        00002360
16 IF ((.NOT. FLAG .AND. TYPE1 .LT. ALPHA).OR.(FLAG .AND. ALPHA .NE. 300002370
+ .AND. TYPE1 .LT. ALPHA)) TYPE1=ALPHA        00002380
C    STRING=HOL.J(-STRING,STRING ""TYPE1"")    00002390
C****    STRING    +    HOL                                00002400
LVVPOS =    J                                00002410
LVVTYP = 3                                00002420
LVFUNC=    HOL                                00002430
LVVARG=    STRING                                00002440
CALL LVFIND(LV2    Q.LV2    R.LV2    S.LV2    T) 00002450
LV1    AAD =    STRING                                00002460
IF (LVVAL.NE.-1) LV1    AAD = LVVAL            00002470
LV1    AAF = LV1    AAD                                00002480
C**** LV1    AAF    -    STRING                                00002490
LVVAD=-1                                00002500
LVVTYP=-1                                00002510
LVVPOS=1                                00002520
LVFUNC=    STRING                                00002530
LVVARG=LV1    AAF                                00002540
CALL LVDLET                                00002550
LV1    AAF = LV1    AAD                                00002560
C**** LV1    AAF    STRING    ""                                00002570
LVDEST= 0                                00002580
LV1    AAK = TYPE1                                00002590
LVTYPE(1) = 1                                00002600
LVVALS(1) = LV1    AAK                                00002610
LVDEST= 0                                00002620
LVVNVL = 1                                00002630
LVFUNC =    STRING                                00002640
LVVARG=LV1    AAF                                00002650
CALL LVNSRT                                00002660
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)            00002670
IF (LVVAL.LT.0) RETURN                        00002680
RETURN                                        00002690
C WILL SCAN AN EXPONENT                                00002700
20 IF (STJ .LT. 0) RETURN                    00002710
C    R+STJ/11 "R                                00002720
C****    R    +    STJ                                00002730
LVVTYP = 3                                00002740
LVVPOS = 1                                00002750

```



LVINDX = 0	00002760
LVFUNC= STJ	00002770
LVVARG= R	00002780
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)	00002790
LV1 AAD = R	00002800
IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00002810
LVVTR = LVVAL	00002820
LVVAL = -100	00002830
IF (LVVTR.EQ.-1) GO TO 11	00002840
C**** LV1 AAD "	00002850
R = LV1 AAD	00002860
FLAG=.TRUE.	00002870
RETURN	00002880
C RECOGNIZED A TERM, PRODUCT OR PRIMARY PERHAPS NEEDING PARENTHEZIZATION	00002890
30 CONTINUE	00002900
KTMP=R	00002910
IF (NDP.EQ. 0) GO TO 34	00002920
LVVPOS=-LVVPOS	00002930
LVVTYP= 3	00002940
LVVPOS= 1	00002950
LVDEST= 2	00002960
LV1 AAM = 1	00002970
LVTYPE(1) = 1	00002980
LVVALS(1) = LV1 AAM	00002990
LVDEST= 2	00003000
LVVAVL = 1	00003010
LVFUNC = FUNC1	00003020
LVVARG= OPRAND	00003030
CALL LVNSRT	00003040
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00003050
IF (LVVAL.LT.0) RETURN	00003060
34 CONTINUE	00003070
ITEST=0	00003080
IF (STJ.LT. 0) GO TO 31	00003090
C R+STJ "R//32	00003100
C**** R + STJ	00003110
LVVTYP = 3	00003120
LVVPOS = 1	00003130
LVINDX = 0	00003140
LVFUNC= STJ	00003150
LVVARG= R	00003160
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)	00003170
LV1 AAF = R	00003180
IF (LVVAL.NE.-1) LV1 AAF = LVVAL	00003190
C**** LV1 AAF "	00003200
R = LV1 AAF	00003210
LVVTR = LVVAL	00003220
LVVAL = -100	00003230
IF (LVVTR.NE.-1) GO TO 32	00003240
C 31 R+STOP/39 "R	00003250
31 CONTINUE	00003260
C**** R + STOP	00003270
LVVTYP = 3	00003280
LVVPOS = 1	00003290
LVINDX = 0	00003300



```

LVFUNC=      STOP                                00003310
LVVARG=      R                                  00003320
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)      00003330
LV1 AAF =    R                                  00003340
IF (LVVAL, NE, -1) LV1 AAF = LVVAL               00003350
LVVTR = LVVAL                                   00003360
LVVAL = -100                                    00003370
IF (LVVTR, EQ, -1) GO TO 39                      00003380
C**** LV1 AAF =    " R                          00003390
      R = LV1 AAF                               00003400
IF (STJ, LT, 0, AND, KTMP, EQ, 889) GO TO 38     00003410
ITEST=-1                                         00003420
32 CONTINUE                                     00003430
C IF UNARY PLUS OR MINUS RETURN                  00003440
IF (STACK(JSTOP,1), NE, 288, AND, STACK(JSTOP,1), NE, 110) GO TO 33 00003450
JLAST=STACK(JSTOP,3)-1                          00003460
IF (STACK(JSTOP,1), EQ, 288) JLAST=JLAST-1       00003470
C STRING+HOL, JLAST(-STRING, STRING, ""TYPE1"") 00003480
C**** STRING * HOL                             00003490
LVVPOS =    JLAST                               00003500
LVVTYP =    3                                   00003510
LVFUNC=    HOL                                 00003520
LVVARG=    STRING                              00003530
CALL LVFIND(LV2 U, LV2 V, LV2 W, LV2 X)         00003540
LV1 AAF =    STRING                            00003550
IF (LVVAL, NE, -1) LV1 AAF = LVVAL               00003560
LV1 AAL = LV1 AAF                              00003570
C**** LV1 AAL - STRING                         00003580
LVVAL=-1                                         00003590
LVVTYP=-1                                        00003600
LVVPOS=1                                         00003610
LVFUNC=    STRING                              00003620
LVVARG=LV1 AAL                                  00003630
CALL LVOLET                                     00003640
LV1 AAL = LV1 AAF                              00003650
C**** LV1 AAL STRING ""                               00003660
LVDEST= 0                                       00003670
LV1 AAM = TYPE1                                00003680
LVTYPE(1) = 1                                  00003690
LVVALS(1) = LV1 AAM                            00003700
LVDEST= 0                                       00003710
LVVNL = 1                                       00003720
LVFUNC=    STRING                              00003730
LVVARG=LV1 AAL                                  00003740
CALL LVNSRT                                     00003750
IF (LVVAL, LT, 0) CALL LVEXIT(LVVAL)             00003760
IF (LVVAL, LT, 0) RETURN                        00003770
C STRING+HOL, J(-STRING, STRING, ""TYPE1"")     00003780
C**** STRING * HOL                             00003790
LVVPOS =    J                                  00003800
LVVTYP =    3                                   00003810
LVFUNC=    HOL                                 00003820
LVVARG=    STRING                              00003830
CALL LVFIND(LV2 Y, LV2 Z, LV2 0, LV2 1)         00003840
LV1 AAF =    STRING                            00003850

```



IF (LVVAL,NE,-1) LV1	AAF = LVVAL	00003860
LV1 AAL = LV1 AAF		00003870
C**** LV1 AAL -	STRING	00003880
LVVAD=-1		00003890
LVVTYP=-1		00003900
LVVPOS=1		00003910
LVFUNC=	STRING	00003920
LVVARG=LV1 AAL		00003930
CALL LVDELETE		00003940
LV1 AAL = LV1 AAF		00003950
C**** LV1 AAL	STRING ""	00003960
LVDEST= 0		00003970
LV1 AAN = TYPE1		00003980
LVTYPE(1) = 1		00003990
LVVALS(1) = LV1 AAN		00004000
LVDEST= 0		00004010
LVVNVL = 1		00004020
LVFUNC =	STRING	00004030
LVVARG=LV1 AAL		00004040
CALL LVNSRT		00004050
IF (LVVAL,LT,0) CALL LVEXIT(LVVAL)		00004060
IF (LVVAL,LT,0) RETURN		00004070
IF (ITEST ,LT, 0) J=J-1		00004080
RETURN		00004090
33 CONTINUE		00004100
IF (ITEST ,LT, 0) J=J-1		00004110
JSTACK=JSTACK+1		00004120
STACK(JSTACK,1)=STOP		00004130
STACK(JSTACK,2)=0		00004140
STACK(JSTACK,3)=J		00004150
STACK(JSTACK,4)=0		00004160
NTMP=R		00004170
CALL SLEVEL(SKIP)		00004180
JSTACK=JSTACK-1		00004190
R=NTMP		00004200
JLAST=1		00004210
IF (JSTOP ,GT, 0) JLAST=STACK(JSTOP,3)		00004220
NTMP=TYPE1		00004230
JJ=JLAST		00004240
IF (STACK(JSTACK,1) ,EQ, 418 ,AND, JLAST ,GT, 1) JJ=JLAST-1		00004250
C STRING+HOL,JJ+STRING "TYPE1		00004260
C**** STRING *	HOL	00004270
LVVPOS =	JJ	00004280
LVVTYP =	3	00004290
LVFUNC=	HOL	00004300
LVVARG=	STRING	00004310
CALL LVFIND(LV2	2,LV2 3,LV2 4,LV2 5)	00004320
LV1 AAF =	STRING	00004330
IF (LVVAL,NE,-1) LV1	AAF = LVVAL	00004340
C**** LV1 AAF *	STRING	00004350
LVVTYP =	3	00004360
LVVPOS =	1	00004370
LVINDX =	0	00004380
LVFUNC=	STRING	00004390
LVVARG= LV1	AAF	00004400



CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)	00004410
LV1 AAL = LV1 AAF	00004420
IF (LVVAL, NE, -1) LV1 AAL = LVVAL	00004430
C**** LV1 AAL " TYPE1	00004440
TYPE1 = LV1 AAL	00004450
IF (.NOT. FLAG .OR. NTMP .EQ. 3) GO TO 35	00004460
IF ((NTMP .EQ. 0 .OR. NTMP .EQ. 2) .AND. (TYPE1 .EQ. 0 .OR. TYPE1 .EQ. 2)) GO TO 35	00004470
ERRFLG = .TRUE.	00004480
CALL ERROR(78, J, IDM2, IDM3, IDM4)	00004490
35 CONTINUE	00004500
IF (TYPE1 .GT. 2 .OR. NTYPE .GT. 1) GO TO 38	00004510
FUNC=FUNC1	00004520
IF (TYPE1 .EQ. 1) FUNC=FUNC2	00004530
IF (TYPE1 .EQ. 2) FUNC=FUNC3	00004540
C STRING+HOL, JLAST LEFT(,1 LPAR,,1 FUNC)	00004550
C**** STRING * HOL	00004560
LVVPOS = JLAST	00004570
LVVTYP = 3	00004580
LVFUNC = HOL	00004590
LVVARG = STRING	00004600
CALL LVFIND(LV2 6, LV2 7, LV2 8, LV2 9)	00004610
LV1 AAL = STRING	00004620
IF (LVVAL, NE, -1) LV1 AAL = LVVAL	00004630
LV1 AAF = LV1 AAL	00004640
LV1 AAO = LEFT	00004650
C**** LV1 AAO . 1	00004660
LVVTYP = 3	00004670
LVVPOS = 1	00004680
C**** LV1 AAF LV1 AAO LPAR	00004690
LVDEST = 1	00004700
LVTYPE(1) = 0	00004710
LVVALS(1) = LPAR	00004720
LVVNVL = 1	00004730
LVFUNC = LV1 AAO	00004740
LVVARG = LV1 AAF	00004750
CALL LVNSRT	00004760
IF (LVVAL, LT, 0) CALL LVEXIT(LVVAL)	00004770
IF (LVVAL, LT, 0) RETURN	00004780
LV1 AAF = LV1 AAL	00004790
C**** LV1 AAO . 1	00004800
LVVTYP = 3	00004810
LVVPOS = 1	00004820
C**** LV1 AAF LV1 AAO FUNC	00004830
LVDEST = 1	00004840
LVTYPE(1) = 0	00004850
LVVALS(1) = FUNC	00004860
LVVNVL = 1	00004870
LVFUNC = LV1 AAO	00004880
LVVARG = LV1 AAF	00004890
CALL LVNSRT	00004900
IF (LVVAL, LT, 0) CALL LVEXIT(LVVAL)	00004910
IF (LVVAL, LT, 0) RETURN	00004920
C STRING+HOL, J RIGHT RPAR	00004930
C**** STRING * HOL	00004940
LVVPOS = J	00004950



LVVTYP = 3	00004960
LVFUNC= HOL	00004970
LVVARG= STRING	00004980
CALL LVFIND(LV2 AA,LV2 AB,LV2 AC,LV2 AD)	00004990
LV1 AAL = STRING	00005000
IF (LVVAL,NE,-1) LV1 AAL = LVVAL	00005010
C**** LV1 AAL RIGHT RPAR	00005020
LVDEST= 0	00005030
LVTYPE(1) = 0	00005040
LVVALS(1) = RPAR	00005050
LVVNL = 1	00005060
LVFUNC = RIGHT	00005070
LVVARG=LV1 AAL	00005080
CALL LVNSRT	00005090
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00005100
IF (LVVAL.LT.0) RETURN	00005110
38 IF (ITEST.LT.0 .AND. STJ.LT.0) J=J+1	00005120
FLAG=.FALSE.	00005130
RETURN	00005140
39 FLAG=.FALSE.	00005150
GO TO 11	00005160
C CHECK FOR CORRECTNESS OF SUBSCRIPTS	00005170
40 NR=R	00005180
C R+STJ/11 "R	00005190
C**** R + STJ	00005200
LVVTYP = 3	00005210
LVVPOS = 1	00005220
LVINDX = 0	00005230
LVFUNC= STJ	00005240
LVVARG= R	00005250
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00005260
LV1 AAL = R	00005270
IF (LVVAL,NE,-1) LV1 AAL = LVVAL	00005280
LVVTR = LVVAL	00005290
LVVAL = -100	00005300
IF (LVVTR,EQ,-1) GO TO 11	00005310
C**** LV1 AAL " R	00005320
R = LV1 AAL	00005330
NBETA=GETDIM(STR(J))	00005340
IF (NR,EQ.359 .AND. NBETA,NE.4) GO TO 47	00005350
ALPHA=GETTYP(STR(J))	00005360
GAMMA=STR(J)/1000000	00005370
IF (NTYPE,EQ.3) GO TO 45	00005380
IF (NR,EQ.839 .AND. NBETA,EQ.0) GO TO 45	00005390
IF (NR,EQ.359) GO TO 45	00005400
IF (NR,EQ.21 .AND. NBETA,EQ.4) GO TO 45	00005410
IF (NBETA,EQ.0 .AND. NR,EQ.935) GO TO 45	00005420
IF (NTYPE,EQ.2 .AND. NR,EQ.935 .AND. STR(J-1),NE,-7) GO TO 45	00005430
IF (NTYPE,EQ.2 .AND. NR,EQ.359 .AND. NBETA,EQ.0) GO TO 11	00005440
CALL ERKOR(79,J,IDM2,IDM3,IDM4)	00005450
ERRFLG=.TRUE.	00005460
45 CONTINUE	00005470
IF (GAMMA,GE.6 .AND. NBETA,EQ.4)	00005480
\$ CALL ERROR(76,IDM1,IDM2,IDM3,IDM4)	00005490
IF (ALPHA,EQ.3) GO TO 46	00005500



N1=ALPHA+1	00005510
ERRFLG=.TRUE.	00005520
CALL ERKOR(80,TYPE(1,N1),TYPE(2,N1),J,DM4)	00005530
46 IF(NBETA .EQ. 4) RETURN	00005540
MARGS=MARGS+1	00005550
LV1 AAO = OPRAND	00005560
LVDEST= 0	00005570
LV1 AAR = MARGS	00005580
LVTYPE(1) = 1	00005590
LVVALS(1) = LV1 AAR	00005600
LVDEST= 0	00005610
LVVNVL = 1	00005620
LVFUNC = FUNC2	00005630
LVVARG=LV1 AAO	00005640
CALL LVNSRT	00005650
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00005660
IF(LVVAL.LT.0) RETURN	00005670
LVDEST= 0	00005680
LV1 AAJ = J	00005690
LVTYPE(1) = 1	00005700
LVVALS(1) = LV1 AAJ	00005710
LVDEST= 0	00005720
LVVNVL = 1	00005730
LVFUNC = FUNC3	00005740
LVVARG=LV1 AAO	00005750
CALL LVNSRT	00005760
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00005770
IF(LVVAL.LT.0) RETURN	00005780
LVDEST= 0	00005790
LV1 AAS = NDEPTH	00005800
LVTYPE(1) = 1	00005810
LVVALS(1) = LV1 AAS	00005820
LVDEST= 0	00005830
LVVNVL = 1	00005840
LVFUNC = LEVEL	00005850
LVVARG=LV1 AAO	00005860
CALL LVNSRT	00005870
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00005880
IF(LVVAL.LT.0) RETURN	00005890
IVR=MARGS	00005900
IF(IVR .GT. 50) GO TO 1610	00005905
ICOL=9	00005910
IVAL=MOD(STR(J),10000)	00005920
IARGS(IVR)=HITPUT(IARGS(IVR),IVAL,ICOL)	00005930
IF(NR .EQ. 839) GO TO 49	00005940
IF(NTYPE .NE. 3) RETURN	00005950
C FLAG SUPSCRIPT IN I/O LIST	00005960
IARGS(IVR)=HITPUT(IARGS(IVR),1,ICOL+2)	00005970
RETURN	00005980
C FLAG DO INDEX IN I/O LIST	00005990
49 IARGS(IVR)=HITPUT(IARGS(IVR),2,ICOL+2)	00006000
IF(NFLAG .LT. 1) RETURN	00006010
IARGS(IVR)=HITPUT(IARGS(IVR),FL(NFLAG),ICOL+7)	00006020
RETURN	00006030
47 NR=NR	00006040



C	SUBSCRIPT DOES NOT BEGIN WITH CONSTANT, FORCE SEARCH FOR VARIABLE	00006050
	GO TO 11	00006060
C	CHECK FOR PROPER NUMBER OF SUBSCRIPTS	00006070
50	IF ( BETA .EQ. 4 .OR. N .NE. 452) GO TO 52	00006080
	MARGS=MARGS+1	00006090
	LV1    AAO =        OPRAND	00006100
	LVDEST= 0	00006110
	LV1    AAT = MARGS	00006120
	LVTYPE(1) = 1	00006130
	LVVALS(1) = LV1    AAT	00006140
	LVDEST= 0	00006150
	LVVNVL = 1	00006160
	LVFUNC =        FUNC2	00006170
	LVVARG=LV1    AAO	00006180
	CALL LVNSRT	00006190
	IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00006200
	IF (LVVAL.LT.0) RETURN	00006210
	LVDEST= 0	00006220
	LV1    AAU = J-1	00006230
	LVTYPE(1) = 1	00006240
	LVVALS(1) = LV1    AAU	00006250
	LVDEST= 0	00006260
	LVVNVL = 1	00006270
	LVFUNC =        FUNC3	00006280
	LVVARG=LV1    AAO	00006290
	CALL LVNSRT	00006300
	IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00006310
	IF (LVVAL.LT.0) RETURN	00006320
	LVDEST= 0	00006330
	LV1    AAV = NDEPTH	00006340
	LVTYPE(1) = 1	00006350
	LVVALS(1) = LV1    AAV	00006360
	LVDEST= 0	00006370
	LVVNVL = 1	00006380
	LVFUNC =        LEVEL	00006390
	LVVARG=LV1    AAO	00006400
	CALL LVNSRT	00006410
	IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00006420
	IF (LVVAL.LT.0) RETURN	00006430
	IF (MARGS .GT. 50) GO TO 1610	00006435
	ICOL=9	00006440
	IVAL=MOD(STR(J-1),10000)	00006450
	IARGS(MARGS)=HITPUT(IARGS(MARGS),IVAL,ICOL)	00006460
	IF (NOPAR .LE. 0) GO TO 52	00006470
	LVVPOS =        1	00006480
	LVVTYP =        3	00006490
	LVVPOS=-LVVPOS	00006500
	LVFUNC=        ACTION	00006510
	LVVARG=        OPRAND	00006520
	CALL LVFIND(LV2    AE,LV2    AF,LV2    AG,LV2    AH)	00006530
	LV1    AAO =        OPRAND	00006540
	IF (LVVAL.NE.-1) LV1    AAO = LVVAL	00006550
	LVVTR = LVVAL	00006560
	LVVAL = -100	00006570
	IF (LVVTR.EQ.-1) GO TO        52	00006580



```

      MFUNC = LV1      AAO
      IARGS(MARGS)=HITPUT(IARGS(MARGS),MFUNC,ICOL*3)
      IARGS(MARGS)=BITPUT(IARGS(MARGS),NARGS,ICOL*9)
C IF NO STRING LEFT, RETURN IF CONSTANT,VARIABLE OR I/O LIST
52 IF(STJ.LT. 0 .AND. (BETA.EQ. 0 .OR. BETA.EQ. 4
   $ .OR. NTYPE.EQ. 3)) RETURN
   IF(BETA.GT. NARRAY) GO TO 55
   ERRFLG=.TRUE.
   CALL ERROR(81,J,IDM2,IDM3,IDM4)
55 IF(R.EQ. 452) NARRAY=0
   IF(R.EQ. 318) NARRAY=1
   IF(R.EQ. 60) NARRAY=2
   IF(R.EQ. 103) NARRAY=3
   IF(STJ.LT. 0) GO TO 58
   LVVTYP = 3
   LVVPOS = 1
   LVVINDX = 0
   LVFUNC=          STJ
   LVVARG=          R
   CALL LVFIND(LVVINDX,LVVINDX,LVVINDX,LVVINDX)
   LV1      AAL =          R
   IF (LVVAL.NE.-1) LV1      AAL = LVVAL
C**** LV1      AAL      "      R
      R = LV1      AAL
   LVVTR = LVVAL
   LVVAL = -100
   IF (LVVTR.NE.-1) GO TO 56
58 IF(NTYPE.EQ. 3 .AND. NARRAY.EQ. 0) GO TO 57
   IF(BETA.GE. 1 .AND. BETA.LE. 3 .AND. NOPAR.EQ. 0)
   $ CALL ERROR(82,J,IDM2,IDM3,IDM4)
57 NARRAY=-1
   GO TO 11
56 IF(NTYPE.EQ. 3 .AND. NARRAY.EQ. 0) RETURN
   IF(STJ.EQ. RPAR .AND. NARRAY.LT. BETA .AND. J.EQ. MAXJ)
   $ CALL ERROR(82,J,IDM2,IDM3,IDM4)
   IF(STJ.EQ. RPAR) NARRAY=-1
   RETURN
C RESET TYPE OF STATEMENT IN ANTICIPATION OF SEARCH FOR BOOLEAN PRIMARY
60 CONTINUE
   NOTFLG=.FALSE.
   IF(STR(J-1).EQ. -17) NOTFLG=.TRUE.
   TYPE1=-1
C STRING+HOL,J(-STRING,STRING ""TYPE1"")
C****      STRING      +      HOL
   LVVPOS =          J
   LVVTYP = 3
   LVFUNC=          HOL
   LVVARG=          STRING
   CALL LVFIND(LV2AI,LV2AJ,LV2AK,LV2AL)
   LV1      AAL =          STRING
   IF (LVVAL.NE.-1) LV1      AAL = LVVAL
   LV1      AAO = LV1      AAL
C**** LV1      AAO      -      STRING
   LVVAD=-1
   LVVTYP=-1

```

```

00006590
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830
00006840
00006850
00006860
00006870
00006880
00006890
00006900
00006910
00006920
00006930
00006940
00006950
00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130

```



LVVPOS=1		00007140
LVFUNC=	STRING	00007150
LVVARG=LV1	AAO	00007160
CALL LVDLET		00007170
LV1	AAO = LV1 AAL	00007180
C****	LV1 AAO STRING ""	00007190
	LVDEST= 0	00007200
	LV1 AAF = TYPE1	00007210
	LVTYPE(1) = 1	00007220
	LVVALS(1) = LV1 AAF	00007230
	LVDEST= 0	00007240
	LVVNL = 1	00007250
	LVFUNC =	00007260
	LVVARG=LV1 AAO	00007270
	CALL LVNSKT	00007280
	IF(LVVAL,LT,0) CALL LVEXIT(LVVAL)	00007290
	IF(LVVAL,LT,0) RETURN	00007300
	IF(STJ,NE,OPRAND) GO TO 65	00007310
	ALPHA=GETTYP(STR(J))	00007320
	BETA=GETDIM(STR(J))	00007330
	IF(ALPHA,NE,4) GO TO 11	00007340
65	CONTINUE	00007350
C	R+STJ/11 "R	00007360
C****	R * STJ	00007370
	LVVTYP = 3	00007380
	LVVPOS = 1	00007390
	LVINDX = 0	00007400
	LVFUNC=	00007410
	LVVARG=	00007420
	CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00007430
	LV1 AAL =	00007440
	IF(LVVAL,NE,-1) LV1 AAL = LVVAL	00007450
	LVVTR = LVVAL	00007460
	LVVAL = -100	00007470
	IF(LVVTR,EQ,-1) GO TO 11	00007480
C****	LV1 AAL " R	00007490
	R = LV1 AAL	00007500
	RETURN	00007510
C	IF BOOLEAN PRIMARY IS AN ARITHMETIC COMPARE CONTINUE PARSING PRIMARY	00007520
70	IF(STJ,LT,0) RETURN	00007530
	IF(TYPE1,EQ,4) GO TO 75	00007540
C	R+STJ "R/11	00007550
C****	R * STJ	00007560
	LVVTYP = 3	00007570
	LVVPOS = 1	00007580
	LVINDX = 0	00007590
	LVFUNC=	00007600
	LVVARG=	00007610
	CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00007620
	LV1 AAL =	00007630
	IF(LVVAL,NE,-1) LV1 AAL = LVVAL	00007640
C****	LV1 AAL " R	00007650
	R = LV1 AAL	00007660
	LVVTR = LVVAL	00007670
	LVVAL = -100	00007680



IF (LVVTR,EQ,-1) GO TO	11	00007690
C RELATIONAL OPERATOR FOUND		00007700
IF (NDEP ,EQ. 0) RETURN		00007710
LVVPOS=LVVPOS		00007720
LVVTYP= 3		00007730
LVVPOS= 1		00007740
LVDEST= 2		00007750
LV1 AAO = 1		00007760
LVTYPE(1) = 1		00007770
LVVALS(1) = LV1 AAO		00007780
LVDEST= 2		00007790
LVVNVL = 1		00007800
LVFUNC = FUNC1		00007810
LVVARG= OPRAND		00007820
CALL LVNSRT		00007830
IF (LVVAL,LT,0) CALL LVEXIT(LVVAL)		00007840
IF (LVVAL,LT,0) RETURN		00007850
C IF BOOLEAN VARIABLE OR CONSTANT, SET STATE TO STOP		00007860
75 R=STOP		00007870
JSTACK=JSTACK+1		00007880
STACK(JSTACK,1)=R		00007890
STACK(JSTACK,2)=0		00007900
STACK(JSTACK,3)=J		00007910
STACK(JSTACK,4)=0		00007920
GO TO 11		00007930
C COMPARE TYPES ON BOTH SIDES OF RELATIONAL EXPRESSION		00007940
80 IF (TYPE1 ,EQ. 0 .OR. TYPE1 ,EQ. 2 .OR. TYPE1 ,EQ. 3) GO TO 85		00007950
ERRFLG=TRUE.		00007960
CALL ERROR(83,J)		00007970
TYPE1=-1		00007980
C STRING+HOL,J(-STRING,STRING "TYPE1")		00007990
C**** STRING + HOL		00008000
LVVPOS = J		00008010
LVVTYP = 3		00008020
LVFUNC= HOL		00008030
LVVARG= STRING		00008040
CALL LVFIND(LV2AM,LV2AN,LV2AO,LV2AP)		00008050
LV1 AAL = STRING		00008060
IF (LVVAL,NE,-1) LV1 AAL = LVVAL		00008070
LV1 AAO = LV1 AAL		00008080
C**** LV1 AAO ~ STRING		00008090
LVVAD=-1		00008100
LVVTYP=-1		00008110
LVVPOS=1		00008120
LVFUNC= STRING		00008130
LVVARG=LV1 AAO		00008140
CALL LVOLET		00008150
LV1 AAO = LV1 AAL		00008160
C**** LV1 AAO STRING ""		00008170
LVDEST= 0		00008180
LV1 AAP = TYPE1		00008190
LVTYPE(1) = 1		00008200
LVVALS(1) = LV1 AAP		00008210
LVDEST= 0		00008220
LVVNVL = 1		00008230



LVFUNC =	STRING	00008240
LVVARG=LV1	AAO	00008250
CALL LVNSRT		00008260
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)		00008270
IF (LVVAL.LT.0) RETURN		00008280
85 TYPE2=TYPE1		00008290
GO TO 11		00008300
C BOOLEAN PRIMARY RECOGNIZED-SET TYPE TO BOOLEAN AND CONTINUE PARSE		00008310
90 IF (TYPE1.EQ. TYPE2 .OR. TYPE1+TYPE2.EQ. 2 .OR.		00008320
+ TYPE2.LT. 0) GO TO 95		00008330
N1=TYPE1+1		00008340
N2=TYPE2+1		00008350
CALL ERROR(77,TYPE(1,N1),TYPE(2,N1),TYPE(1,N2),TYPE(2,N2))		00008360
ERRFLG=.TRUE.		00008370
95 TYPE1=4		00008380
TYPE2=-1		00008390
IF (STJ.LT. 0) RETURN		00008400
C STRING+HOL,J(-STRING,STRING ""TYPE1"")		00008410
C**** STRING + HOL		00008420
LVVPOS = J		00008430
LVVTYP = 3		00008440
LVFUNC= HOL		00008450
LVVARG= STRING		00008460
CALL LVFIND(LV2AQ,LV2AR,LV2AS,LV2AT)		00008470
LV1 AAL = STRING		00008480
IF (LVVAL.NE.-1) LV1 AAL = LVVAL		00008490
LV1 AAO = LV1 AAL		00008500
C**** LV1 AAO - STRING		00008510
LVVAD=-1		00008520
LVVTYP=-1		00008530
LVVPOS=1		00008540
LVFUNC= STRING		00008550
LVVARG=LV1 AAO		00008560
CALL LVDLET		00008570
LV1 AAO = LV1 AAL		00008580
C**** LV1 AAO STRING ""		00008590
LVDEST= 0		00008600
LV1 AAQ = TYPE1		00008610
LVTYPE(1) = 1		00008620
LVVALS(1) = LV1 AAQ		00008630
LVDEST= 0		00008640
LVVNVL = 1		00008650
LVFUNC = STRING		00008660
LVVARG=LV1 AAO		00008670
CALL LVNSRT		00008680
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)		00008690
IF (LVVAL.LT.0) RETURN		00008700
GO TO 11		00008710
C PARSE REACHED BLIND ALLEY-MUST BACK UP AND REMOVE PARENTHESES CREATED		00008720
999 JM=STACK(JSTACK,3)		00008730
K=JM		00008740
DO 996 KK=JM,J		00008750
C STRING+HOL,K/995+STRING,1/996 "TYPE1		00008760
C**** STRING + HOL		00008770
LVVPOS = K		00008780



LVVTYP = 3	00008790
LVFUNC= HOL	00008800
LVVARG= STRING	00008810
CALL LVFIND(LV2AU,LV2AV,LV2AW,LV2AX)	00008820
LV1 AAL = STRING	00008830
IF (LVVAL.NE,-1) LV1 AAL = LVVAL	00008840
LVVTH = LVVAL	00008850
LVVAL = -100	00008860
IF (LVVTR.EQ,-1) GO TO 995	00008870
C**** LV1 AAL + STRING	00008880
LVVPOS = 1	00008890
LVVTYP = 3	00008900
LVFUNC= STRING	00008910
LVVARG= LV1 AAL	00008920
CALL LVFIND(LV2AY,LV2AZ,LV2A0,LV2A1)	00008930
LV1 AAO = LV1 AAL	00008940
IF (LVVAL.NE,-1) LV1 AAO = LVVAL	00008950
LVVTR = LVVAL	00008960
LVVAL = -100	00008970
IF (LVVTR.EQ,-1) GO TO 996	00008980
C**** LV1 AAO " TYPE1	00008990
TYPE1 = LV1 AAO	00009000
GO TO 995	00009010
996 K=K-1	00009020
995 CONTINUE	00009030
DO 998 I=JM,J	00009040
C STRING+HOL,I(-LEFT,-RIGHT)	00009050
C**** STRING + HOL	00009060
LVVPOS = 1	00009070
LVVTYP = 3	00009080
LVFUNC= HOL	00009090
LVVARG= STRING	00009100
CALL LVFIND(LV2A2,LV2A3,LV2A4,LV2A5)	00009110
LV1 AAO = STRING	00009120
IF (LVVAL.NE,-1) LV1 AAO = LVVAL	00009130
LV1 AAL = LV1 AAO	00009140
C**** LV1 AAL - LEFT	00009150
LVVAD=-1	00009160
LVVTYP=-1	00009170
LVVPOS=1	00009180
LVFUNC= LEFT	00009190
LVVARG=LV1 AAL	00009200
CALL LVDLET	00009210
LV1 AAL = LV1 AAO	00009220
C**** LV1 AAL - RIGHT	00009230
LVVAD=-1	00009240
LVVTYP=-1	00009250
LVVPOS=1	00009260
LVFUNC= RIGHT	00009270
LVVARG=LV1 AAL	00009280
CALL LVDLET	00009290
998 CONTINUE	00009300
980 CONTINUE	00009310
LVVPOS = 1	00009320
LVVTYP = 3	00009330



LVVPOS=-LVVPOS	00009340
LVFUNC= FUNC3	00009350
LVVARG= OPRAND	00009360
CALL LVFIND(LV2 A6,LV2 A7,LV2 A8,LV2 A9)	00009370
LV1 AAY = OPRAND	00009380
IF (LVVAL.NE.-1) LV1 AAY = LVVAL	00009390
JN = LV1 AAY	00009400
IF(JN .LT. JM) GO TO 985	00009410
LV1 AAY = OPRAND	00009420
LVVPOS = 1	00009430
LVVTYP = 3	00009440
LVVPOS=-LVVPOS	00009450
LVFUNC= FUNC2	00009460
LVVARG= LV1 AAY	00009470
LVVAD=-1	00009480
CALL LVDLET	00009490
LVVPOS = 1	00009500
LVVTYP = 3	00009510
LVVPOS=-LVVPOS	00009520
LVFUNC= FUNC3	00009530
LVVARG= LV1 AAY	00009540
LVVAD=-1	00009550
CALL LVDLET	00009560
LVVPOS = 1	00009570
LVVTYP = 3	00009580
LVVPOS=-LVVPOS	00009590
LVFUNC= LEVEL	00009600
LVVARG= LV1 AAY	00009610
LVVAD=-1	00009620
CALL LVDLET	00009630
GO TO 980	00009640
985 CONTINUE	00009650
LVVPOS = 1	00009660
LVVTYP = 3	00009670
LVVPOS=-LVVPOS	00009680
LVFUNC= FUNC2	00009690
LVVARG= OPRAND	00009700
CALL LVFIND(LV2 BA,LV2 BB,LV2 BC,LV2 BD)	00009710
LV1 AAY = OPRAND	00009720
IF (LVVAL.NE.-1) LV1 AAY = LVVAL	00009730
MARG5 = LV1 AAY	00009740
LVVPOS = 1	00009750
LVVTYP = 3	00009760
LVVPOS=-LVVPOS	00009770
LVFUNC= LEVEL	00009780
LVVARG= OPRAND	00009790
CALL LVFIND(LV2 BE,LV2 BF,LV2 BG,LV2 BH)	00009800
LV1 AAY = OPRAND	00009810
IF (LVVAL.NE.-1) LV1 AAY = LVVAL	00009820
NDEPTH = LV1 AAY	00009830
RETURN	00009840
	00009850
C RECOGNIZED FUNCTION-PREPARE TO SET TYPE OF ARGUMENTS FOR THE "NDEPTH"	00009860
C FUNCTION IN THIS STMT	00009870
1000 CONTINUE	00009880



NDEPTH=NDEPTH+1	00009890
NDEP=NDEP+1	00009900
NARGS=0	00009910
C R*STJ/11 "R	00009920
C**** R * STJ	00009930
LVVTYP = 3	00009940
LVVPOS = 1	00009950
LVINDX = 0	00009960
LVFUNC= STJ	00009970
LVVARG= R	00009980
CALL LVFIND(LVINDX, LVINDX, LVINDX, LVINDX)	00009990
LV1 AAO = R	00010000
IF (LVVAL, NE, -1) LV1 AAO = LVVAL	00010010
LVVTR = LVVAL	00010020
LVVAL = -100	00010030
IF (LVVTR, EQ, -1) GO TO 11	00010040
C**** LV1 AAO " R	00010050
R = LV1 AAO	00010060
NARGS=1	00010070
C OPRAND(OPRAND ""TYPE1"", STRING""NARGS"", ACTION ""NDEPTH"")	00010080
LV1 AAO = OPRAND	00010090
C**** LV1 AAO OPRAND ""	00010100
LVDEST= 0	00010110
LV1 AAL = TYPE1	00010120
LVTYPE(1) = 1	00010130
LVVALS(1) = LV1 AAL	00010140
LVDEST= 0	00010150
LVVNL = 1	00010160
LVFUNC = OPRAND	00010170
LVVARG=LV1 AAO	00010180
CALL LVNSRT	00010190
IF (LVVAL, LT, 0) CALL LVEXIT(LVVAL)	00010200
IF (LVVAL, LT, 0) RETURN	00010210
C**** LV1 AAO STRING ""	00010220
LVDEST= 0	00010230
LV1 AAR = NARGS	00010240
LVTYPE(1) = 1	00010250
LVVALS(1) = LV1 AAR	00010260
LVDEST= 0	00010270
LVVNL = 1	00010280
LVFUNC = STRING	00010290
LVVARG=LV1 AAO	00010300
CALL LVNSRT	00010310
IF (LVVAL, LT, 0) CALL LVEXIT(LVVAL)	00010320
IF (LVVAL, LT, 0) RETURN	00010330
C**** LV1 AAO ACTION ""	00010340
LVDEST= 0	00010350
LV1 AAS = NDEPTH	00010360
LVTYPE(1) = 1	00010370
LVVALS(1) = LV1 AAS	00010380
LVDEST= 0	00010390
LVVNL = 1	00010400
LVFUNC = ACTION	00010410
LVVARG=LV1 AAO	00010420
CALL LVNSRT	00010430



IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00010440
IF (LVVAL.LT.0) RETURN	00010450
C OPRAND FUNC1 ""J+1""	00010460
C**** OPRAND FUNC1 ""	00010470
LVDEST= 0	00010480
LVIAAO=0	00010490
LVTYPE(1) = 1	00010500
LVVALS(1) = LV1 AAO	00010510
LVDEST= 0	00010520
LVVNVL = 1	00010530
LVFUNC = FUNC1	00010540
LVVARG= OPRAND	00010550
CALL LVNSRT	00010560
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)	00010570
IF (LVVAL.LT.0) RETURN	00010580
TYPE1=-1	00010590
C NOPAR=NOPAR+1	00010600
NOPAR=NOPAR+1	00010610
RETURN	00010620
	00010630
C KEEP TRACK OF THE NUMBER AND TYPES OF ARGUMENTS IN FUNCTION CALLS	00010640
C MUST USE STACK FOR POSSIBLE RECURSIVE FUNCTION USE	00010650
1100 CONTINUE	00010660
LVVTYP = 3	00010670
LVVPOS = 1	00010680
LVINDX = 0	00010690
LVFUNC= STJ	00010700
LVVARG= R	00010710
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00010720
LVIAAT=R	00010730
IF (LVVAL.NE.-1) LVIAAT=LVVAL	00010740
LVVTR = LVVAL	00010750
LVVAL = -100	00010760
IF (LVVTR.EQ.-1) GO TO 11	00010770
R=LVIAAT	00010780
LVIAAT=OPRAND	00010790
LVVPOS = 1	00010800
LVVTYP = 3	00010810
LVVPOS=-LVVPOS	00010820
LVFUNC= STRING	00010830
LVVARG= LV1 AAT	00010840
CALL LVFIND(LV2HI,LV2HI,LV2HI,LV2HI)	00010850
LV1 AAU = LV1 AAT	00010860
IF (LVVAL.NE.-1) LV1 AAU = LVVAL	00010870
LVVTR = LVVAL	00010880
LVVAL = -100	00010890
IF (LVVTR.EQ.-1) GO TO 1103	00010900
C**** LV1 AAU " NARGS	00010910
NARGS = LV1 AAU	00010920
C**** LV1 AAT + STRING	00010930
LVVPOS = 1	00010940
LVVTYP = 3	00010950
LVVPOS=-LVVPOS	00010960
LVFUNC= STRING	00010970
LVVARG= LV1 AAT	00010980



LVVAD=-1	00010990
CALL LVDLET	00011000
C**** LV1 AAT * ACTION	00011010
LVVPOS = 1	00011020
LVVTYP = 3	00011030
LVVPOS=-LVVPOS	00011040
LVFUNC= ACTION	00011050
LVVARG= LV1 AAT	00011060
CALL LVFIND(LV2BM, LV2BN, LV2BO, LV2BP)	00011070
LV1 AAU = LV1 AAT	00011080
IF (LVVAL, NE, -1) LV1 AAU = LVVAL	00011090
MFUNC=LV1AAU	00011100
LVVPOS = 1	00011110
LVVTYP = 3	00011120
LVVPOS=-LVVPOS	00011130
LVFUNC= FUNC1	00011140
LVVARG= OPRAND	00011150
CALL LVFIND(LV2 BQ, LV2 BR, LV2 BS, LV2 BT)	00011160
LV1 AA3 = OPRAND	00011170
IF (LVVAL, NE, -1) LV1 AA3 = LVVAL	00011180
IEXP = LV1 AA3	00011190
1103 CONTINUE	00011200
C STORE ARGUMENT TYPES	00011210
IF (NDEPTH, GT, 5) CALL ERROR(85, IDM1, IDM2, IDM3, IDM4)	00011220
IF (NDEPTH, GT, 5) GO TO 1130	00011230
IF (NARGS, LE, 63) GO TO 1104	00011240
ERRFLG=.TRUE.	00011250
CALL ERROR(84, NDEPTH, IDM2, IDM3, IDM4)	00011260
GO TO 11	00011270
1104 CONTINUE	00011280
MM=(5+NARGS)/3	00011290
ITEMP=NARGS-3*(MM-2)	00011300
ICOL=9*ITEMP-6	00011310
NARY(MFUNC, MM)=BITPUT(NARY(MFUNC, MM), TYPE1+1, ICOL)	00011320
IF (STR(J-2), NE, -6 .AND. STR(J-2), NE, -4) GO TO 1130	00011330
NDIM=GETDIM(STR(J-1))	00011340
IF (NDIM, GE, 4) GO TO 1130	00011350
C STORE DIMENSIONALITY OF ARGUMENTS	00011360
NARY(MFUNC, MM)=BITPUT(NARY(MFUNC, MM), NDIM, ICOL+3)	00011370
1130 CONTINUE	00011380
NARY(MFUNC, MM)=BITPUT(NARY(MFUNC, MM), IEXP, ICOL+6)	00011390
IF (STJ, EQ, COMMA) GO TO 1105	00011400
NARY(MFUNC, 1)=NARGS	00011410
LV1 AA3 = OPRAND	00011420
LVVPOS = 1	00011430
LVVTYP = 3	00011440
LVVPOS=-LVVPOS	00011450
LVFUNC= ACTION	00011460
LVVARG= LV1 AA3	00011470
LVVAD=-1	00011480
CALL LVDLET	00011490
LVVPOS = 1	00011500
LVVTYP = 3	00011510
LVVPOS=-LVVPOS	00011520
LVFUNC= OPRAND	00011530



LVVARG= LV1	AA3					00011540
CALL LVFIND(LV2	BU, LV2	BV, LV2	BW, LV2	BX)		00011550
LV1	AA4 = LV1	AA3				00011560
IF (LVVAL, NE, -1)	LV1	AA4 = LVVAL				00011570
LVVTR = LVVAL						00011580
LVVAL = -100						00011590
IF (LVVTR, EQ, -1)	GO TO	1135				00011600
TYPE1 = LV1	AA4					00011610
LV1	AA3 =	OPRAND				00011620
LVVPOS =	1					00011630
LVVTYP =	3					00011640
LVVPOS=-LVVPOS						00011650
LVFUNC=	FUNC1					00011660
LVVARG= LV1	AA3					00011670
LVVAD=-1						00011680
CALL LVDMLET						00011690
LVVPOS =	1					00011700
LVVTYP =	3					00011710
LVVPOS=-LVVPOS						00011720
LVFUNC=	OPRAND					00011730
LVVARG= LV1	AA3					00011740
LVVAD=-1						00011750
CALL LVDMLET						00011760
1135 NOPAR=NOPAR-1						00011770
NDEP=NDEP-1						00011780
RETURN						00011790
1105 TYPE1=-1						00011800
C STRING=HOL, J(-STRING, STRING	""TYPE1""")					00011810
C****	STRING	+	HOL			00011820
LVVPOS =	J					00011830
LVVTYP=3						00011840
LVFUNC=	HOL					00011850
LVVARG=	STRING					00011860
CALL LVFIND(LV2HY, LV2HZ, LV2H0, LV2H1)						00011870
LV1	AAT =	STRING				00011880
IF (LVVAL, NE, -1)	LV1	AAT = LVVAL				00011890
LV1	AAU = LV1	AAT				00011900
C****	LV1	AAU	-	STRING		00011910
LVVAD=-1						00011920
LVVTYP=-1						00011930
LVVPOS=1						00011940
LVFUNC=	STRING					00011950
LVVARG=LV1	AAU					00011960
CALL LVDMLET						00011970
LV1	AAU = LV1	AAT				00011980
C****	LV1	AAU	STRING	""		00011990
LVDEST=	0					00012000
LV1	AAV = TYPE1					00012010
LVTYPE(1) =	1					00012020
LVVALS(1) =	LV1	AAV				00012030
LVDEST=	0					00012040
LVNVNL =	1					00012050
LVFUNC =	STRING					00012060
LVVARG=LV1	AAU					00012070
CALL LVNSRT						00012080



IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00012090
IF(LVVAL.LT.0) RETURN	00012100
NARGS=NARGS+1	00012110
C OPRAND(STRING ""NARGS"",FUNC1 ,-1 ""J+1"")	00012120
C**** LV1 AAT STRING ""	00012130
LVDEST= 0	00012140
LV1 AAU = NARGS	00012150
LVTYPE(1) = 1	00012160
LVVALS(1) = LV1 AAU	00012170
LVDEST= 0	00012180
LVVNL = 1	00012190
LVFUNC = STRING	00012200
LVVARG=OPRAND	00012210
CALL LVNSRT	00012220
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00012230
IF(LVVAL.LT.0) RETURN	00012240
C**** FUNC1 *- 1	00012250
LVVPOS=-LVVPOS	00012260
LVVTYP= 3	00012270
LVVPOS= 1	00012280
C**** LV1 AAT FUNC1 ""	00012290
LVDEST= 2	00012300
LVIAAW=0	00012310
LVTYPE(1) = 1	00012320
LVVALS(1) = LV1 AAU	00012330
LVDEST= 2	00012340
LVVNL = 1	00012350
LVFUNC = FUNC1	00012360
LVVARG= OPRAND	00012370
CALL LVNSRT	00012380
IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)	00012390
IF(LVVAL.LT.0) RETURN	00012400
RETURN	00012410
C SAVE TYPE OF STATEMENT WHILE PARSING EXPONENT	00012420
1200 CONTINUE	00012430
C STRING+HOL.J(-STRING,STRING ""TYPE1"")	00012440
C**** STRING + HOL	00012450
LVVPOS = J	00012460
LVVTYP = 3	00012470
LVFUNC= HOL	00012480
LVVARG= STRING	00012490
CALL LVFIND(LV2H2,LV2H3,LV2H4,LV2H5)	00012500
LV1 AAT = STRING	00012510
IF (LVVAL.NE.-1) LV1 AAT = LVVAL	00012520
LV1 AAX = LV1 AAT	00012530
C**** LV1 AAX - STRING	00012540
LVVAD=-1	00012550
LVVTYP=-1	00012560
LVVPOS=1	00012570
LVFUNC= STRING	00012580
LVVARG=LV1 AAX	00012590
CALL LVULET	00012600
LV1 AAX = LV1 AAT	00012610
C**** LV1 AAX STRING ""	00012620
LVDEST= 0	00012630



LV1	AA1 = TYPE1	00012640
LVTYPE(1)	= 1	00012650
LVVALS(1)	= LV1 AA1	00012660
LVDEST	= 0	00012670
LVVNL	= 1	00012680
LVFUNC	= STRING	00012690
LVVARG=LV1	AA1	00012700
CALL LVNSRT		00012710
IF (LVVAL.LT.0)	CALL LVEXIT(LVVAL)	00012720
IF (LVVAL.LT.0)	RETURN	00012730
TYPE1=-1		00012740
C	R+STJ/11 "R	00012750
C****	R + STJ	00012760
LVVTYP	= 3	00012770
LVVPOS	= 1	00012780
LVINDX	= 0	00012790
LVFUNC	= STJ	00012800
LVVARG	= R	00012810
CALL LVFIND(LVINDX, LVINDX, LVINDX)		00012820
LV1	AA1 = R	00012830
IF (LVVAL.NE.-1)	LV1 AA1 = LVVAL	00012840
LVVTR	= LVVAL	00012850
LVVAL	= -100	00012860
IF (LVVTR.EQ.-1)	GO TO 11	00012870
C****	LV1 AA1 " R	00012880
	R = LV1 AA1	00012890
C	COMPLETE	00012900
	RETURN	00012910
1300	CONTINUE	00012920
	IF (STJ.LT. 0) RETURN	00012930
	IF (STJ.NE. AND .AND. STJ.NE. OR .AND. STJ.NE. NOT) GO TO 11	00012940
LVVTYP	= 3	00012950
LVVPOS	= 1	00012960
LVINDX	= 0	00012970
LVFUNC	= STJ	00012980
LVVARG	= R	00012990
CALL LVFIND(LVINDX, LVINDX, LVINDX)		00013000
LV1	AA6 = R	00013010
IF (LVVAL.NE.-1)	LV1 AA6 = LVVAL	00013020
LVVTR	= LVVAL	00013030
LVVAL	= -100	00013040
IF (LVVTR.EQ.-1)	GO TO 11	00013050
	R = LV1 AA6	00013060
	IF (NDP.EQ. 0) RETURN	00013070
C	LOGICAL OPERATOR FOUND	00013080
	LVVPOS=-LVVPOS	00013090
LVVTYP	= 3	00013100
LVVPOS	= 1	00013110
LVDEST	= 2	00013120
LV1	AA6 = 1	00013130
LVTYPE(1)	= 1	00013140
LVVALS(1)	= LV1 AA6	00013150
LVDEST	= 2	00013160
LVVNL	= 1	00013170
LVFUNC	= FUNC1	00013180



LVVARG=	OPRAND	00013190
CALL LVNSRT		00013200
IF (LVVAL.LT.0) CALL LVEXIT(LVVAL)		00013210
IF (LVVAL.LT.0) RETURN		00013220
RETURN		00013230
1400 CONTINUE		00013240
LVVTYP = 3		00013250
LVVPOS = 1		00013260
LVINDX = 0		00013270
LVFUNC=	STJ	00013280
LVVARG=	R	00013290
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)		00013300
LV1 AA7 =	R	00013310
IF (LVVAL.NE.-1) LV1	AA7 = LVVAL	00013320
LVVTR = LVVAL		00013330
LVVAL = -100		00013340
IF (LVVTR.EQ.-1) GO TO	11	00013350
R = LV1	AA7	00013360
C LEFT PAREN FOUND IN I/O LIST		00013370
NFLAG=NFLAG+1		00013380
FL(NFLAG)=MARGS		00013390
RETURN		00013400
1500 CONTINUE		00013410
LVVTYP = 3		00013420
LVVPOS = 1		00013430
LVINDX = 0		00013440
LVFUNC=	STJ	00013450
LVVARG=	R	00013460
CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)		00013470
LV1 AA7 =	R	00013480
IF (LVVAL.NE.-1) LV1	AA7 = LVVAL	00013490
LVVTR = LVVAL		00013500
LVVAL = -100		00013510
IF (LVVTR.EQ.-1) GO TO	11	00013520
R = LV1	AA7	00013530
IF (STJ.EQ.COMMA) RETURN		00013540
C RIGHT PAREN FOUND IN I/O LIST		00013550
NFLAG=NFLAG-1		00013560
RETURN		00013570
1600 CONTINUE		00013580
PRINT 1601		00013590
1601 FORMAT(1X,10H ACTION 16)		00013600
TYPE1=4		00013610
RETURN		00013620
1610 CALL ERROR(95,IDM1,IDM2,IDM3,IDM4)		00013622
STOP		00013624
25000 CONTINUE		00013630
LV2A=0		00013640
LV2B=0		00013650
LV2C=0		00013660
LV2D=0		00013670
LV2E=0		00013680
LV2F=0		00013690
LV2G=0		00013700
LV2H=0		00013710



LV2I=0  
 LV2J=0  
 LV2K=0  
 LV2L=0  
 LV2M=0  
 LV2N=0  
 LV2O=0  
 LV2P=0  
 LV2Q=0  
 LV2R=0  
 LV2S=0  
 LV2T=0  
 LV2U=0  
 LV2V=0  
 LV2W=0  
 LV2X=0  
 LV2Y=0  
 LV2Z=0  
 LV20=0  
 LV21=0  
 LV22=0  
 LV23=0  
 LV24=0  
 LV25=0  
 LV26=0  
 LV27=0  
 LV28=0  
 LV29=0  
 LV2AA=0  
 LV2AB=0  
 LV2AC=0  
 LV2AD=0  
 LV2AE=0  
 LV2AF=0  
 LV2AG=0  
 LV2AH=0  
 LV2AI=0  
 LV2AJ=0  
 LV2AK=0  
 LV2AL=0  
 LV2AM=0  
 LV2AN=0  
 LV2AO=0  
 LV2AP=0  
 LV2AQ=0  
 LV2AR=0  
 LV2AS=0  
 LV2AT=0  
 LV2AU=0  
 LV2AV=0  
 LV2AW=0  
 LV2AX=0  
 LV2AY=0  
 LV2AZ=0  
 LV2A0=0

00013720  
 00013730  
 00013740  
 00013750  
 00013760  
 00013770  
 00013780  
 00013790  
 00013800  
 00013810  
 00013820  
 00013830  
 00013840  
 00013850  
 00013860  
 00013870  
 00013880  
 00013890  
 00013900  
 00013910  
 00013920  
 00013930  
 00013940  
 00013950  
 00013960  
 00013970  
 00013980  
 00013990  
 00014000  
 00014010  
 00014020  
 00014030  
 00014040  
 00014050  
 00014060  
 00014070  
 00014080  
 00014090  
 00014100  
 00014110  
 00014120  
 00014130  
 00014140  
 00014150  
 00014160  
 00014170  
 00014180  
 00014190  
 00014200  
 00014210  
 00014220  
 00014230  
 00014240  
 00014250  
 00014260



LV2A1=0  
LV2A2=0  
LV2A3=0  
LV2A4=0  
LV2A5=0  
LV2A6=0  
LV2A7=0  
LV2A8=0  
LV2A9=0  
LV2BA=0  
LV2BB=0  
LV2BC=0  
LV2BD=0  
LV2BE=0  
LV2BF=0  
LV2BG=0  
LV2BH=0  
LV2BI=0  
LV2BJ=0  
LV2BK=0  
LV2BL=0  
LV2BM=0  
LV2BN=0  
LV2BO=0  
LV2BP=0  
LV2BQ=0  
LV2BR=0  
LV2BS=0  
LV2BT=0  
LV2BU=0  
LV2BV=0  
LV2BW=0  
LV2BX=0  
LV2BY=0  
LV2BZ=0  
LV2C1=0  
LV2C2=0  
LV2C3=0  
LV2C4=0  
LV2C5=0  
GO TO 25001  
END

00014270  
00014280  
00014290  
00014300  
00014310  
00014320  
00014330  
00014340  
00014350  
00014360  
00014370  
00014380  
00014390  
00014400  
00014410  
00014420  
00014430  
00014440  
00014450  
00014460  
00014470  
00014480  
00014490  
00014500  
00014510  
00014520  
00014530  
00014540  
00014550  
00014560  
00014570  
00014580  
00014590  
00014600  
00014610  
00014620  
00014630  
00014640  
00014650  
00014660  
00014670  
00014680



SUBROUTINE SEPAR	00000010
COMMON A(1326),U(500),IDTHL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKOT,MODE,IERR,IDES	00000040
COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,	00000050
1 DIR,ICOM,ISEP	00000060
INTEGER A,SEPST,SEPND,DIR,BLANK,SLASH,COMMA	00000070
DATA BLANK/'H ',SLASH/'H',//,COMMA/'H,/'	00000080
ICOM=0	00000090
ISLASH=0	00000100
DO 20 I=1,N	00000110
JJ=SEPST*DIR*(I-1)	00000120
IF(A(JJ).EQ. BLANK) GO TO 20	00000130
IF(A(JJ).EQ. SLASH) GO TO 5	00000140
IF(A(JJ).EQ. COMMA) GO TO 10	00000150
GO TO 30	00000160
5 CONTINUE	00000170
ISLASH=1	00000180
IF(ICOM.EQ. 1) GO TO 40	00000190
GO TO 20	00000200
10 IF(ISLASH.EQ. 1.OR. ICOM.EQ. 1) GO TO 40	00000210
ICOM=1	00000220
20 CONTINUE	00000230
GO TO 40	00000240
30 IF(ISLASH.EQ. 0.AND. ICOM.EQ. 0) GO TO 35	00000250
ISEP=1	00000260
SEPND=JJ-DIR	00000270
RETURN	00000280
35 CONTINUE	00000290
ISEP=0	00000300
SEPND=JJ	00000310
RETURN	00000320
40 ISEP=-1	00000330
RETURN	00000340
END	00000350



SUBROUTINE SIMP	00000010
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGIU,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES	00000040
COMMON/HASHLK/IBLOCK(2500),NBLOCK,NB,NBRNCH	00000050
DIMENSION IALPH1(7),IALPH2(4),IALPH3(5),IALPH4(10)	00000060
DATA IALPH1/1HP,1HE,1HT,1HU,1HP,1HN,1M /	00000070
DATA IALPH2/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE,1M /	00000080
DATA IALPH3/1HS,1HT,1HO,1HP,1H /	00000090
DATA IALPH4/1HR,1HL,1HO,1HC,1HK,1HD,1HA,1HT,1HA,1H /	00000100
IF(ITYP.EQ. 10) GO TO 25	00000110
IF(ITYP.EQ. 7) GO TO 15	00000120
IF(ITYP.EQ. 29) GO TO 35	00000130
DO 10 I=1,7	00000140
IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50	00000150
10 CONTINUE	00000160
NR=1	00000170
NBRNCH=1	00000180
NBLOCK=NBLOCK+1	00000190
IBLOCK(NBLOCK)=999	00000200
RETURN	00000210
15 DO 20 I=1,9	00000220
IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50	00000230
20 CONTINUE	00000240
RETURN	00000250
25 DO 30 I=1,5	00000260
IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50	00000270
30 CONTINUE	00000280
NR=1	00000290
NBRNCH=1	00000300
NBLOCK=NBLOCK+1	00000310
IBLOCK(NBLOCK)=999	00000320
RETURN	00000330
35 DO 40 I=1,10	00000340
IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 50	00000350
40 CONTINUE	00000360
RETURN	00000370
50 CALL ERROR(7,IUM1,IUM2,IUM3,IUM4)	00000380
RETURN	00000390
END	00000400



	SUBROUTINE SLEVEL(FAIL)	00000010
	COMMON/LVARG/LVFUNC,LVARG,LVVAD,LVVPOS,LVVTP,LVVAL,	00000020
	+LVHEAD,LVVNL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
	COMMON/LVTABL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSQSP( 1)	00000040
	COMMON /NEED/ START,ASSOC,LEVEL,STOP	00000050
	COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)	00000060
	COMMON /STRING/ NNN(2),STR	00000070
	COMMON /JL/ JSTOP	00000080
	INTEGER START,STOP,ASSOC,STACK,STR(500),STJ,R,RTEMP	00000090
	LOGICAL FAIL	00000100
C	EXECUTE	00000110
	GO TO 25000	00000120
25001	CONTINUE	00000130
	RTEMP=0	00000140
	JSTOP=JSTACK	00000150
10	IF(JSTOP.EQ. 0) GO TO 40	00000160
	NPNTR=STACK(JSTOP,4)	00000170
	IF(NPNTR.GT. 0) GO TO 20	00000180
	IF(STACK(JSTOP,2).NE. 0) GO TO 30	00000190
	JSTOP=JSTOP-1	00000200
	GO TO 10	00000210
20	JSTOP=NPNTR-1	00000220
	GO TO 10	00000230
30	STACK(JSTACK,4)=JSTOP	00000240
	JAS=STACK(JSTOP,2)	00000250
	R=STACK(JSTOP,1)	00000260
	RTEMP=R	00000270
C	R=LEVEL,JAS "R	00000280
C****	R + LEVEL	00000290
	LVVPOS = JAS	00000300
	LVVTP = 3	00000310
	LVFUNC= LEVEL	00000320
	LVVARG= R	00000330
	CALL LVFIND(LV2 A,LV2 B,LV2 C,LV2 D)	00000340
	LV1 AAD = R	00000350
	IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00000360
C****	LV1 AAD " R	00000370
	R = LV1 AAD	00000380
	FAIL=.FALSE.	00000390
	IF(STACK(JSTOP,4).LT. 0) STACK(JSTOP,4)=0	00000400
	RETURN	00000410
40	FAIL=.TRUE.	00000420
	RETURN	00000430
C	COMPLETE	00000440
25000	CONTINUE	00000450
	LV2A=0	00000460
	LV2H=0	00000470
	LV2C=0	00000480
	LV2D=0	00000490
	GO TO 25001	00000500
	END	00000510



SUBROUTINE SQUEEZ	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
INTEGER A,D,BLANK,AICH	00000050
DATA BLANK/1H /,AICH/1HH/	00000060
J=0	00000070
DO 10 I=1,M	00000080
IF(D(I) .EQ. BLANK) GO TO 10	00000090
J=J+1	00000100
D(J)=D(I)	00000110
IF(D(J) .NE. AICH) GO TO 10	00000120
IF(JTYP .NE. 3) GO TO 10	00000130
M=M+J-1	00000140
RETURN	00000150
10 CONTINUE	00000160
M=J	00000170
RETURN	00000180
END	00000190



	SUBROUTINE SSTOP(FAIL)	00000010
	COMMON/LVARG/LVFUNC,LVARG,LVVAD,LVVPOS,LVVTP,LVVAL,	00000020
	+LVHEAD,LVVNL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP	00000030
	COMMON/LVTAHL/LVTSIZ,LVMAP( 1)/LVVSEQ/LVSIZE,LVSQSP( 1)	00000040
	COMMON /NEED/ START,ASSOC,LEVEL,STOP	00000050
	COMMON/NEEDS/STJ,JSTACK,R,JAS,JLAST,RTEMP,STACK(400,4)	00000060
	COMMON /STRING/ NNN(2),STR	00000070
	INTEGER START,STOP,ASSOC,STACK,STR(1),STJ,R,TEMP	00000080
	LOGICAL FAIL	00000090
C	EXECUTE	00000100
	GO TO 25000	00000110
25001	CONTINUE	00000120
	JSTOPS=JSTACK	00000130
C 5	R=ASSOC/10	00000140
5	CONTINUE	00000150
C****	R + ASSOC	00000160
	LVVTP = 3	00000170
	LVVPOS = 1	00000180
	LVINDX = 0	00000190
	LVFUNC= ASSOC	00000200
	LVVARG= R	00000210
	CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00000220
	LV1 AAD = R	00000230
	IF (LVVAL.NE.-1) LV1 AAD = LVVAL	00000240
	LVVTR = LVVAL	00000250
	LVVAL = -100	00000260
	IF (LVVTR.EQ.-1) GO TO 10	00000270
	JSTOPS=JSTOPS+1	00000280
	STACK(JSTOPS,1)=R	00000290
	STACK(JSTOPS,2)=0	00000300
	STACK(JSTOPS,4)=0	00000310
C 10	R=(STOP//20,+STOP/30 "R/5/5)	00000320
10	CONTINUE	00000330
	LV1 AAD = R	00000340
C****	LV1 AAD = STOP	00000350
	LVVAL = -100	00000360
	IF (LV1 AAD.NE. STOP) LVVAL = -1	00000370
	LVVTR = LVVAL	00000380
	LVVAL = -100	00000390
	IF (LVVTR.NE.-1) GO TO 20	00000400
C****	LV1 AAD + STOP	00000410
	LVVTP = 3	00000420
	LVVPOS = 1	00000430
	LVINDX = 0	00000440
	LVFUNC= STOP	00000450
	LVVARG= LV1 AAD	00000460
	CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)	00000470
	LV1 AAF = LV1 AAD	00000480
	IF (LVVAL.NE.-1) LV1 AAF = LVVAL	00000490
	LVVTR = LVVAL	00000500
	LVVAL = -100	00000510
	IF (LVVTR.EQ.-1) GO TO 30	00000520
C****	LV1 AAF " R	00000530
	R = LV1 AAF	00000540
	LVVTR = LVVAL	00000550



LVVAL = -100		00000560
IF (LVVTR.EQ.-1) GO TO	5	00000570
IF (LVVTR.NE.-1) GO TO	5	00000580
20 FAIL=.FALSE.		00000590
RETURN		00000600
C 30 JSTACK=JSTOPS/40		00000610
30 CONTINUE		00000620
C**** JSTACK = JSTOPS		00000630
LVVAL = -100		00000640
IF ( JSTACK.NE. JSTOPS) LVVAL = -1		00000650
LVVTR = LVVAL		00000660
LVVAL = -100		00000670
IF (LVVTR.EQ.-1) GO TO	40	00000680
FAIL=.TRUE.		00000690
RETURN		00000700
40 R=STACK(JSTOPS,1)		00000710
JAS=STACK(JSTOPS,2)+1		00000720
R+ASSOC.JAS "TEMP//50		00000730
C**** R + ASSOC		00000740
LVVPOS = JAS		00000750
LVVTR = 3		00000760
LVFUNC= ASSOC		00000770
LVVARG= H		00000780
CALL LVFIND(LV2 A,LV2 B,LV2 C,LV2 D)		00000790
LV1 AAD = R		00000800
IF (LVVAL.NE.-1) LV1 AAD = LVVAL		00000810
C**** LV1 AAD " TEMP		00000820
TEMP = LV1 AAD		00000830
LVVTR = LVVAL		00000840
LVVAL = -100		00000850
IF (LVVTR.NE.-1) GO TO	50	00000860
JSTOPS=JSTOPS-1		00000870
C //30		00000880
LVVTR = LVVAL		00000890
LVVAL = -100		00000900
IF (LVVTR.NE.-1) GO TO	30	00000910
50 STACK(JSTOPS,2)=JAS		00000920
C R=TEMP//40		00000930
C**** R = TEMP		00000940
LVVAL = -100		00000950
IF ( R.NE. TEMP) LVVAL = -1		00000960
LVVTR = LVVAL		00000970
LVVAL = -100		00000980
IF (LVVTR.NE.-1) GO TO	40	00000990
C TEMP "R//5		00001000
C**** TEMP " R		00001010
R = TEMP		00001020
LVVTR = LVVAL		00001030
LVVAL = -100		00001040
IF (LVVTR.NE.-1) GO TO	5	00001050
C COMPLETE		00001060
RETURN		00001070
25000 CONTINUE		00001080
LV2A=0		00001090
LV2B=0		00001100
LV2C=0		00001110
LV2D=0		00001120
GO TO 25001		00001130
END		00001140



SUBROUTINE STATNO	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES	00000040
COMMON/LABELS/STATRA(2,200),NLABEL	00000050
COMMON/DOLLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFLW	00000060
COMMON/HASHBK/IRLOCK(2500),NBLOCK,NB,NBRNCH	00000070
INTEGER A,HLANK,STATRA	00000080
INTEGER BITPUT,BITGET	00000090
DATA HLANK/1H /	00000100
LOC=0	00000110
DO 5 I=1,5	00000120
IF(A(I) .NE. HLANK) GO TO 10	00000130
5 CONTINUE	00000140
IF(ITYP .NE. 18) GO TO 7	00000150
IF(IBLKDT .EQ. 1) RETURN	00000160
IF(NBRNCH .EQ. 0) GO TO 110	00000170
IRLOCK(IBLKST)=BITPUT(IRLOCK(IBLKST),NBRNCH,6)	00000180
RETURN	00000190
7 IF(ITYP .EQ. 28) GO TO 90	00000200
IF(NBLOCK .EQ. 0) GO TO 8	00000210
IF(IRLOCK(NBLOCK) .EQ. 998) GO TO 32	00000220
IF(NH .EQ. 2) GO TO 31	00000230
IF(NH .EQ. 1) GO TO 70	00000240
RETURN	00000250
8 NBLOCK=1	00000260
GO TO 34	00000270
10 IF(ITYP .EQ. 18) GO TO 50	00000280
JPTR=1	00000290
CALL GNLE	00000300
IF(JTYP .NE. 5) GO TO 50	00000310
IF(A(6) .NE. HLANK) GO TO 50	00000320
IF(JPTR .LT. 6) GO TO 50	00000330
CALL STSRCH	00000340
IF(BITGET(STATRA(2,LOC),9,3) .EQ. 1) GO TO 60	00000350
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,9)	00000360
IF(LTYP .EQ. 9) GO TO 20	00000370
STATRA(2,LOC)=BITPUT(STATRA(2,LOC),ITYP,6)	00000380
GO TO 30	00000390
20 STATRA(2,LOC)=BITPUT(STATRA(2,LOC),9,6)	00000400
30 IF(ITYP .EQ. 28) RETURN	00000410
IF(NBLOCK .EQ. 0) GO TO 8	00000420
IF(NH .EQ. 1) GO TO 32	00000430
31 NBLOCK=NBLOCK+1	00000440
IRLOCK(NBLOCK)=998	00000450
NBRNCH=1	00000460
32 NH=0	00000470
NBLOCK=NBLOCK+1	00000480
IRLOCK(IBLKST)=BITPUT(IRLOCK(IBLKST),NBLOCK,24)	00000490
IRLOCK(IBLKST)=BITPUT(IRLOCK(IBLKST),NBRNCH,6)	00000500
J1=IBLKST+1	00000510
J2=NBLOCK-NBRNCH-1	00000520
IF(IRLOCK(J2) .GT. 6000) GO TO 34	00000530
J21=J2-1	00000540
21 IF(IRLOCK(J1) .LT. 6000) GO TO 34	00000550



IRES=IBLOCK(J1)	00000560
DO 22 K2=J1,J21	00000570
22 IBLOCK(K2)=IHLOCK(K2+1)	00000580
IBLOCK(J2)=IRES	00000590
GO TO 21	00000600
34 IHLKST=NBLOCK	00000610
NBRNCH=0	00000620
IBLOCK(IHLKST)=HITPUT(0,ILOOP,12)	00000630
IF(LOC.EQ.0) RETURN	00000640
IBLOCK(IHLKST)=HITPUT(IHLOCK(IHLKST),LOC,32)	00000650
STATRA(2,LOC)=HITPUT(STATRA(2,LOC),IHLKST,32)	00000660
IF(RITGET(STATRA(2,LOC),15,3).NE.1) RETURN	00000670
IF(IOVFLW.EQ.1) RETURN	00000680
IF(LOC.NE.ISTACK(1,ILOOP)) GO TO 80	00000690
IF(ITYP.GE.3.AND.ITYP.LE.6) GO TO 100	00000700
IF(ITYP.EQ.9.OR.ITYP.EQ.10.OR.ITYP.EQ.17) GO TO 100	00000710
NH=2	00000720
ISTACK(2,ILOOP)=1	00000730
NBLOCK=NBLOCK+1	00000740
IBLOCK(NBLOCK)=6000+ISTACK(4,ILOOP)	00000750
KLOOP=ILOOP-1	00000760
DO 40 J=1,KLOOP	00000770
LOOP=ILOOP-J	00000780
IF(ISTACK(2,LOOP).EQ.1) GO TO 40	00000790
IF(ISTACK(1,LOOP).EQ.LOC) GO TO 35	00000800
ILOOP=LOOP	00000810
RETURN	00000820
35 ISTACK(2,LOOP)=1	00000830
NBLOCK=NBLOCK+1	00000840
IBLOCK(NBLOCK)=6000+ISTACK(4,LOOP)	00000850
40 CONTINUE	00000860
ILOOP=0	00000870
RETURN	00000880
50 IERC=32	00000890
GO TO 200	00000900
60 IERC=33	00000910
GO TO 200	00000920
70 IERC=34	00000930
GO TO 200	00000940
80 IERC=35	00000950
GO TO 200	00000960
90 IERC=36	00000970
GO TO 200	00000980
100 IERC=37	00000990
GO TO 200	00010000
110 IERC=38	00010010
200 CALL ERROR(IERC,IDM1,IDM2,IDM3,IDM4)	00010020
RETURN	00010030
END	00010040



SUBROUTINE STFNC(NFNC)	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
*LOC,LTYP,ITYP,IPLKDT,MODE,IERR,IDES	00000040
COMMON/FUNC/IFNCRA(5,22),MARGS,IARGS(50),FNCLOC(5),NFUNC	00000050
INTEGER BITGET	00000060
NARG=BITGET(IDTBL(3,LOC),7,6)	00000070
NAR2=IFNCRA(NFNC,1)	00000080
IF (NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,LOC),IDTBL(2,LOC),	00000090
NARGS=MIN0(NARG,NAR2)	00000100
KOUNT=0	00000110
NT=1+(NARG-1)/3	00000120
DO 10 I=1,NT	00000130
ICOL1=-6	00000140
ICOL2=-3	00000150
DO 10 J=1,3	00000160
KOUNT=KOUNT+1	00000170
IF (KOUNT .GT. NARGS) RETURN	00000180
ICOL1=ICOL1+9	00000190
ICOL2=ICOL2+9	00000200
IF (BITGET(IFNCRA(NFNC,I+1),ICOL2,3) .NE. 0)	00000210
* CALL ERROR(50,KOUNT,I1,I2,I3)	00000220
ITP1=BITGET(IFNCRA(NFNC,I+1),ICOL1,3)	00000230
IF (ITP1 .EQ. 0) GO TO 10	00000240
ITP2=BITGET(IDTBL(3,LOC+KOUNT),10,3)	00000250
IF (ITP1 .NE. ITP2) CALL ERROR(51,KOUNT,I1,I2,I3)	00000260
10 CONTINUE	00000270
RETURN	00000280
END	00000290



```

SUBROUTINE STORE
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISWCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
NID=NID+1
IF(NID.GT. 500) GO TO 20
IF(INITID(IDTYP).NE. 0) GO TO 5
INITID(IDTYP)=NID
5 CONTINUE
IDTBL(1,NID)=NXTID(1)
IDTBL(2,NID)=NXTID(2)
IDTBL(4,NID)=0
IF(LASTID(IDTYP).EQ. 0) GO TO 10
LAST=LASTID(IDTYP)
IDTBL(4,LAST)=NID
10 LASTID(IDTYP)=NID
RETURN
20 WRITE(6,25)
25 FORMAT(/////5X,46H SYMBOL TABLE OVERFLOW - PROCESSING TERMINATED)
STOP
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210

```

```

SUBROUTINE STSRCH
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISWCH(3),
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES
COMMON/LABELS/STATRA(2,200),NLABEL
INTEGER STATRA
IF(NLABEL.EQ. 0) GO TO 15
DO 10 I=1,NLABEL
IF(STATRA(1,I).NE. N2) GO TO 10
LOC=I
RETURN
10 CONTINUE
15 NLABEL=NLABEL+1
IF(NLABEL.GT. 200) GO TO 20
LOC=NLABEL
STATRA(1,LOC)=N2
RETURN
20 WRITE(6,25)
25 FORMAT(/////5X,53H STATEMENT NO. TABLE OVERFLOW - PROCESSING TERMINATED)
* NATED)
STOP
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220

```



SUBROUTINE S11H	00000010
COMMON A(132),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
• NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
• LOC,LTYP,ITYP,IBLNDT,MODE,IEPR,IDES	00000040
DIMENSION IALPH(10),IALPH2(8),KT(5)	00000050
INTEGER BLANK,COMMA,RPAR,A,D	00000060
INTEGER HITPUT	00000070
DATA IALPH/1HS,1HU,1HB,1HR,1HU,1HU,1HT,1HI,1HN,1HE/	00000080
DATA IALPH2/1HF,1HU,1HN,1HC,1HT,1HI,1HU,1HN/	00000090
DATA KT/1HR,1HC,1HU,1HI,1HL/	00000100
DATA BLANK/1H //,LPAR/1H(/,RPAR/1H)/,COMMA/1H,/	00000110
DATA IF/1HF/	00000120
ISTATE=0	00000130
NARG=0	00000140
IF(ITYP, EQ, 30) GO TO 5	00000150
2 DO 3 I=1,8	00000160
IF(NEXT(JPTR), NE, IALPH2(I)) GO TO 50	00000170
3 CONTINUE	00000180
GO TO 12	00000190
5 DO 10 I=1,10	00000200
IF(NEXT(JPTH), NE, IALPH(I)) GO TO 50	00000210
10 CONTINUE	00000220
GO TO 17	00000230
12 JPTR=JPTR	00000240
IST=1	00000250
IFIRST=NEXT(IST)	00000260
IF(IFIRST, EQ, IF) GO TO 14	00000270
DO 13 I=1,5	00000280
IF(IFIRST, NE, KT(I)) GO TO 13	00000290
ISTATE=I	00000300
GO TO 14	00000310
13 CONTINUE	00000320
14 JPTR=JPTR	00000330
17 CALL GNLE	00000340
IF(JTYP, NE, 2) GO TO 50	00000350
IDTYP=2	00000360
CALL STORE	00000370
IE=D(1)	00000380
IF(ITYP, NE, 31) GO TO 15	00000390
IF(NEXT(JPTR), NE, LPAR) GO TO 50	00000400
IFNCNM=NID	00000410
GO TO 20	00000420
15 IF(NEXT(JPTR), EQ, BLANK) GO TO 30	00000430
IF(A(JPTR-1), NE, LPAR) GO TO 50	00000440
20 CALL GNLE	00000450
IF(JTYP, NE, 2) GO TO 60	00000460
CALL SEARCH	00000470
IF(ISRCH(1), NE, 0, OR, ISRCH(2), NE, 0)	00000480
\$ CALL ERROR(R6,NXTID(1),NXTID(2),IDM2,IDM3)	00000490
IDTYP=1	00000500
CALL STORE	00000510
IDTBL(3,NID)=HITPUT(IDTBL(3,NID),1,12)	00000520
NARG=NARG+1	00000530
IF(NEXT(JPTR), EQ, RPAR) GO TO 30	00000540
IF(A(JPTR-1), NE, COMMA) GO TO 50	00000550



GO TO 20	00000560
30 LOC#1	00000570
D(1)=IE	00000580
IF(NARG,GT, 63) CALL ERROR(83, IDM1, IDM2, IDM3, IDM4)	00000590
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NARG,7)	00000600
IF(ITYP, EQ, 30) RETURN	00000610
IF(ISTATE, EQ, 0) GO TO 55	00000620
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)	00000630
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)	00000640
RETURN	00000650
55 CALL IMPTYP	00000660
RETURN	00000670
50 CALL ERROR(7, IDM1, IDM2, IDM3, IDM4)	00000680
RETURN	00000690
60 CALL ERROR(86, NXTID(1), NXTID(2), IDM3, IDM4)	00000700
RETURN	00000710
END	00000720



```

SUBROUTINE SUBCHK                                00000010
COMMON A(1326),U(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IOTYP,NID, 00000030
* LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES 00000040
COMMON/GLOBAL/NHLK,NREF,NSUBS,BLKTB(200),EXTTBL(100),ISUBS(100) 00000050
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600) 00000060
INTEGER BITPUT,BITGET 00000070
NSUBS=NSUBS+1 00000080
IF(NSUBS.GT.100) GO TO 50 00000090
NARG=BITGET(IDTBL(3,1),7,6) 00000100
ITP=BITGET(IDTBL(3,1),10,3) 00000110
IF(BITGET(IDTBL(3,1),18,1).EQ.1) GO TO 15 00000120
IDTBL(3,1)=BITPUT(IDTBL(3,1),1,18) 00000130
DO 5 I=1,NLIST 00000140
IF(IDTBL(1,I).NE.ISUBLT(1,I)) GO TO 5 00000150
IF(IDTBL(2,I).NE.ISUBLT(2,I)) GO TO 5 00000160
LISTLC=I 00000170
IDTBL(3,1)=BITPUT(IDTBL(3,1),LISTLC,32) 00000180
GO TO 20 00000190
5 CONTINUE 00000200
CALL ERROR(52,IDM1,IDM2,IDM3,IDM4) 00000210
NLIST=NLIST+1 00000220
ISUBLT(1,NLIST)=IDTBL(1,1) 00000230
ISUBLT(2,NLIST)=IDTBL(2,1) 00000240
ISUBS(NSUBS)=NLIST 00000250
ISUBLT(3,NLIST)=0 00000260
ISUBLT(4,NLIST)=0 00000270
IDTBL(3,1)=BITPUT(IDTBL(3,1),NLIST,32) 00000280
IF(NARG.EQ.0) RETURN 00000290
IPTR=NINTFC+1 00000300
ISUBLT(3,NLIST)=BITPUT(0,NARG,6) 00000310
ISUBLT(3,NLIST)=BITPUT(ISUBLT(3,NLIST),ITP,13) 00000320
ISUBLT(4,NLIST)=IPTR 00000330
NINTFC=IPTR+(NARG-1)/3 00000340
KOUNT=0 00000350
DO 10 I=IPTR,NINTFC 00000360
INTFAC(I)=0 00000370
ICOL1=-6 00000380
ICOL2=-3 00000390
DO 10 J=1,3 00000400
KOUNT=KOUNT+1 00000410
IF(KOUNT.GT.NARG) RETURN 00000420
ICOL1=ICOL1+9 00000430
ICOL2=ICOL2+9 00000440
ITP=BITGET(IDTBL(3,KOUNT+1),10,3) 00000450
NDIM=BITGET(IDTBL(3,KOUNT+1),7,6) 00000460
IDTBL(3,KOUNT+1)=BITPUT(IDTBL(3,KOUNT+1),1,15) 00000470
INTFAC(I)=BITPUT(INTFAC(I),ITP,ICOL1) 00000480
10 INTFAC(I)=BITPUT(INTFAC(I),NDIM,ICOL2) 00000490
RETURN 00000500
15 LISTLC=BITGET(IDTBL(3,1),32,9) 00000510
20 ISUBS(NSUBS)=LISTLC 00000520
KLAS=BITGET(ISUBLT(3,LISTLC),10,4) 00000530
NAR2=BITGET(ISUBLT(3,LISTLC),6,6) 00000540
IF(NARG.NE.NAR2) CALL ERROR(26,IDTBL(1,1),IDTBL(2,1),IDM3,IDM4) 00000550

```



NARGS=MIN0(NARG,NAR2)	00000560
IF (ITP .NE. HITGET(ISUHLT(3,LISTLC),13,3))	00000570
5 CALL ERROR(49,IDTBL(1,1),IDTBL(2,1),IDM3,IDM4)	00000580
IF (NARGS .EQ. 0) RETURN	00000590
IPTR=ISUHLT(4,LISTLC)	00000600
NDPTR=IPTR+(NARGS-1)/3	00000610
KOUNT=0	00000620
DO 25 I=IPTR,NDPTR	00000630
ICOL1=-6	00000640
ICOL2=-3	00000650
DO 25 J=1,3	00000660
KOUNT=KOUNT+1	00000670
IF (KOUNT .GT. NARGS) RETURN	00000680
ICOL1=ICOL1+9	00000690
ICOL2=ICOL2+9	00000700
ITP=HITGET(INTFAC(I),ICOL1,3)	00000710
NDIM=HITGET(INTFAC(I),ICOL2,3)	00000720
ITP2=HITGET(IDTBL(3,KOUNT+1),10,3)	00000730
NDIM2=HITGET(IDTBL(3,KOUNT+1),7,6)	00000740
IOSTAT=HITGET(INTFAC(I),ICOL2+2,2)	00000750
IF (IOSTAT .EQ. 2 .OR. KLAS .EQ. 0) IOSTAT=1	00000760
IDTBL(3,KOUNT+1)=HITPUT(IDTBL(3,KOUNT+1),IOSTAT,15)	00000770
IF (NDIM .NE. NDIM2) CALL ERROR(50,KOUNT,IDM2,IDM3,IDM4)	00000780
IF (ITP2 .NE. 0) GO TO 23	00000790
ITP2=1	00000800
IFST=HITGET(IDTBL(1,KOUNT+1),8,8)	00000810
IF (IFST .LE. 213 .AND. IFST .GE. 201) ITP2=4	00000820
23 IF (ITP .NE. ITP2) CALL ERROR(51,KOUNT,IDM2,IDM3,IDM4)	00000830
25 CONTINUE	00000840
RETURN	00000850
50 CALL ERROR(25,IDM1,IDM2,IDM3,IDM4)	00000860
STOP	00000870
END	00000880

SUBROUTINE SWITCH	00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3),	00000020
* NxtID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID,	00000030
* LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IDES	00000040
DO 20 I=1,NID	00000050
IF (IDTBL(4,I) .NE. LOC) GO TO 20	00000060
IDTBL(4,I)=IDTBL(4,LOC)	00000070
IF (LASTID(1) .EQ. LOC) LASTID(1)=I	00000080
GO TO 30	00000090
20 CONTINUE	00000100
INITID(1)=IDTBL(4,LOC)	00000110
30 LAST=LASTID(2)	00000120
IDTBL(4,LAST)=LOC	00000130
IDTBL(4,LOC)=0	00000140
LASTID(2)=LOC	00000150
CALL ERROR(87,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)	00000160
RETURN	00000170
END	00000180



```

SUBROUTINE SYMTAB                                00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000030
* LOC,LTYP,ITYP,IHLKDT,MODE,IERR,IUES 00000040
COMMON/LABELS/STATRA(2,200),NLABEL 00000050
COMMON/LIST/NLIST,NINTFC,ISUBLT(4,200),INTFAC(600) 00000060
COMMON/STFUNC/NSTFNC,ISTFNC(10) 00000070
DIMENSION ITABEL(9),IRA(2) 00000080
INTEGER TYPE(2,5),DIMS(3),BITGET,STATRA 00000090
DATA TYPE/4HREAL,1H ,4HCOMP,3HLEX,4HDOUB, 00000100
* 2HLE,4HINTE,3HGER,4HLOGI,3HCAL/ 00000110
DATA IBLANK/1H /,IRA(1)/4HARRA/,IRA(2)/1HY/,IFP/4HF,P./, 00000120
* ISLASH/2H/// 00000130
DATA DIMS/1H1,1H2,1H3/ 00000140
IF(NID .LE. 1) RETURN 00000150
IF(IHLKDT .EQ. 1) GO TO 2 00000160
INTL=INITID(2) 00000170
WRITE(6,1) IDTBL(1,INTL),IDTBL(2,INTL) 00000180
1 FORMAT(/////45X,25H SYMBOL TABLE FOR MODULE ,A4,A2) 00000190
GO TO 4 00000200
2 WRITE(6,3) 00000210
3 FORMAT(/////46X,28H SYMBOL TABLE FOR BLOCK DATA) 00000220
4 LOC=INITID(1) 00000230
IF(LOC .EQ. 0) GO TO 28 00000240
WRITE(6,5) 00000250
5 FORMAT(/56X,9H VARIABLES/30X,4HNAME,12X,4HTYPE,31X,10HRELOCATION) 00000260
100 ITABEL(1)=IDTBL(1,LOC) 00000270
ITABEL(2)=IDTBL(2,LOC) 00000280
IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 0) GO TO 27 00000290
I=BITGET(IDTBL(3,LOC),10,3) 00000300
ITABEL(3)=TYPE(1,I) 00000310
ITABEL(4)=TYPE(2,I) 00000320
IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 16 00000330
ITABEL(5)=IRA(1) 00000340
ITABEL(6)=IRA(2) 00000350
I=BITGET(IDTBL(3,LOC),7,6) 00000360
ITABEL(7)=DIMS(I) 00000370
GO TO 18 00000380
16 ITABEL(5)=IBLANK 00000390
ITABEL(6)=IBLANK 00000400
ITABEL(7)=IBLANK 00000410
18 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 0) GO TO 20 00000420
ITABEL(8)=IFP 00000430
ITABEL(9)=IBLANK 00000440
GO TO 25 00000450
20 IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1) GO TO 22 00000460
ITABEL(8)=IBLANK 00000470
ITABEL(9)=IBLANK 00000480
GO TO 25 00000490
22 ICOMNM=IDTBL(9,LOC) 00000500
ITABEL(8)=IDTBL(1,ICOMNM) 00000510
ITABEL(9)=IDTBL(2,ICOMNM) 00000520
IF(ITABEL(8) .EQ. IBLANK) ITABEL(8)=ISLASH 00000530
25 WRITE(6,26) (ITABEL(I),I=1,9) 00000540
26 FORMAT(30X,A4,A2,11X,A4,A3,14X,A4,A1,1X,A1,7X,A4,A2) 00000550

```



27 LOC=IDTBL(4,LOC)	00000560
IF(LOC,NE,0) GO TO 100	00000570
28 IF(1BLKDT,EQ,1) GO TO 60	00000580
LOC=IDTBL(4,INTL)	00000590
IF(LOC,EQ,0) GO TO 60	00000600
WRITE(6,31)	00000610
31 FORMAT(/55X,10H EXTERNALS/44X,4HNAME,10X,4HTYPE,10X,4HARGS)	00000620
30 ITABEL(1)=IDTBL(1,LOC)	00000630
ITABEL(2)=IDTBL(2,LOC)	00000640
LISTLC=BITGET(IDTBL(3,LOC),32,9)	00000650
IF(LISTLC,EQ,0) GO TO 39	00000660
ITP=BITGET(ISUBLT(3,LISTLC),13,3)	00000670
IF(ITP,EQ,0) GO TO 32	00000680
ITABEL(3)=TYPE(1,ITP)	00000690
ITABEL(4)=TYPE(2,ITP)	00000700
GO TO 35	00000710
32 ITABEL(3)=IBLANK	00000720
ITABEL(4)=IBLANK	00000730
35 IF(BITGET(ISUBLT(3,LISTLC),14,1),EQ,1) GO TO 37	00000740
ITABEL(5)=BITGET(ISUBLT(3,LISTLC),6,6)	00000750
WRITE(6,36) (ITABEL(I),I=1,5)	00000760
36 FORMAT(44X,A4,A2,8X,A4,A3,8X,I2)	00000770
GO TO 39	00000780
37 WRITE(6,38) (ITABEL(I),I=1,4)	00000790
38 FORMAT(44X,A4,A2,8X,A4,A3,8X,2H>1)	00000800
39 LOC=IDTBL(4,LOC)	00000810
IF(LOC,NE,0) GO TO 30	00000820
60 IF(NSTFNC,EQ,0) GO TO 40	00000830
WRITE(6,62)	00000840
62 FORMAT(/50X,20H STATEMENT FUNCTIONS/	00000850
\$ 44X,4HNAME,10X,4HTYPE,10X,4HARGS)	00000860
DO 70 I=1,NSTFNC	00000870
LC=ISTFNC(I)	00000880
ITP=BITGET(IDTBL(3,LC),10,3)	00000890
NRG=BITGET(IDTBL(3,LC),7,6)	00000900
70 WRITE(6,36) IDTBL(1,LC),IDTBL(2,LC),TYPE(1,ITP),TYPE(2,ITP),NRG	00000910
40 IF(NLABEL,EQ,0) GO TO 50	00000920
WRITE(6,42)	00000930
42 FORMAT(/51X,17H STATEMENT LABELS)	00000940
WRITE(6,45) (STATRA(1,I),I=1,NLABEL)	00000950
45 FORMAT(40X,5I8)	00000960
DO 47 I=1,NLABEL	00000970
IF(BITGET(STATRA(2,I),9,3),NE,1)	00000980
\$ CALL ERROR(15,STATRA(1,I),IDM2,IDM3,IDM4)	00000990
IF(BITGET(STATRA(2,I),12,3),NE,1)	00001000
\$ CALL ERROR(16,STATRA(1,I),IDM2,IDM3,IDM4)	00001010
47 CONTINUE	00001020
50 LOC=INITID(3)	00001030
IF(LOC,EQ,0) RETURN	00001040
WRITE(6,52)	00001050
52 FORMAT(/53X,14H COMMON BLOCKS/50X,4HNAME,10X,6HLENGTH)	00001060
51 ITABEL(1)=IDTBL(1,LOC)	00001070
IF(ITABEL(1),EQ,IBLANK) ITABEL(1)=ISLASH	00001080
ITABEL(2)=IDTBL(2,LOC)	00001090
WRITE(6,55) ITABEL(1),ITABEL(2),IDTBL(5,LOC)	00001100
55 FORMAT(50X,A4,A2,8X,I8)	00001110
LOC=IDTBL(4,LOC)	00001120
IF(LOC,NE,0) GO TO 51	00001130
RETURN	00001140
END	00001150



```

SUBROUTINE TYPE                                00000010
COMMON A(1326),D(500),IDTBL(11,500),INITID(3),LASTID(3),ISRCH(3), 00000020
* NXTID(2),JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,IDTYP,NID, 00000030
* LOC,LTYP,ITYP,IBLKDT,MODE,IERR,IDES 00000040
DIMENSION IALPH1(7),IALPH2(7),IALPH3(4),IALPH4(7),IALPH5(15), 00000050
1 IDIM(3) 00000060
INTEGER A,RPAR,COMMA,BLANK 00000070
INTEGER BITPUT,BITGET,COMLOC 00000080
DATA IALPH1/1HL,1HO,1HG,1HI,1HC,1HA,1HL/ 00000090
DATA IALPH2/1HI,1HN,1HT,1HE,1HG,1HE,1HR/ 00000100
DATA IALPH3/1HP,1HE,1HA,1HL/ 00000110
DATA IALPH4/1HC,1HO,1HM,1HP,1HL,1HE,1HX/ 00000120
DATA IALPH5/1HD,1HO,1HU,1HB,1HL,1HE,1HP,1HR,1HE,1HC,1HI,1HS,1HI, 00000130
* 1HO,1HN/ 00000140
DATA LPAR/1H(/,RPAR/1H/),COMMA/1H/,BLANK/1H / 00000150
MUL=1 00000160
IT=ITYP-18 00000170
GO TO (10,20,30,40,50),IT 00000180
10 DO 15 I=1,7 00000190
IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 110 00000200
15 CONTINUE 00000210
ISTATE=4 00000220
GO TO 60 00000230
20 DO 25 I=1,4 00000240
IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 110 00000250
25 CONTINUE 00000260
ISTATE=1 00000270
GO TO 60 00000280
30 DO 35 I=1,15 00000290
IF(NEXT(JPTR) .NE. IALPH5(I)) GO TO 110 00000300
35 CONTINUE 00000310
MUL=2 00000320
ISTATE=3 00000330
GO TO 60 00000340
40 DO 45 I=1,7 00000350
IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 110 00000360
45 CONTINUE 00000370
MUL=2 00000380
ISTATE=2 00000390
GO TO 60 00000400
50 DO 55 I=1,7 00000410
IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 110 00000420
55 CONTINUE 00000430
ISTATE=5 00000440
60 ISUB=0 00000450
INCR=MUL 00000460
CALL GNLE 00000470
IF(JTYP .NE. 2) GO TO 110 00000480
CALL SEARCH 00000490
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4) 00000500
IF(ISRCH(1) .EQ. 1) GO TO 62 00000510
IDTYP=1 00000520
CALL STORE 00000530
LOC=NID 00000540
62 IF(BITGET(IDTBL(3,LOC),11,1) .NE. 0) GO TO 120 00000550

```



IF(NEXT(JPTR) .NE. LPAR) GO TO 87	00000560
ISUB=1	00000570
IE=LOC	00000580
I=0	00000590
68 I=I+1	00000600
CALL GNLE	00000610
IF(JTYP .NE. 5) GO TO 65	00000620
IDIM(I)=N2	00000630
IF(N2 .GT. 2**17-1) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)	00000640
IF(N2 .LE. 0) CALL ERROR(8,IDM1,IDM2,IDM3,IDM4)	00000650
INCR=INCR*N2	00000660
GO TO 67	00000670
65 IF(JTYP .NE. 2) GO TO 110	00000680
CALL SEARCH	00000690
IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID(1),NXTID(2),IDM3,IDM4)	00000700
IF(ISRCH(1) .EQ. 1) GO TO 66	00000710
IDTYP=1	00000720
CALL STORE	00000730
LOC=NI0	00000740
66 IF(BITGET(IDTBL(3,LOC),12,1) .NE. 1)	00000750
\$ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)	00000760
IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 130	00000770
CALL IMPTYP	00000780
IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4)	00000790
\$ CALL ERROR(9,IDM1,IDM2,IDM3,IDM4)	00000800
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,13)	00000810
IDIM(I)=2**17+LOC	00000820
67 IF(NEXT(JPTR) .EQ. COMMA) GO TO 68	00000830
IF(A(JPTR-1) .NE. RPAR) GO TO 110	00000840
K=NEXT(JPTR)	00000850
LOC=IE	00000860
IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)	00000870
\$ CALL ERROR(11,IDTBL(1,LOC),IDTBL(2,LOC),IDM3,IDM4)	00000880
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)	00000890
IF(I .GT. 3) GO TO 110	00000900
DO 80 J=1,I	00000910
80 IDTBL(4+J,LOC)=IDIM(J)	00000920
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)	00000930
87 IF(INCR .EQ. 1) GO TO 90	00000940
IF(BITGET(IDTBL(3,LOC),16,1) .NE. 1) GO TO 90	00000950
COMLOC=IDTBL(9,LOC)	00000960
IDTBL(5,COMLOC)=IDTBL(5,COMLOC)+INCR-1	00000970
90 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)	00000980
IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)	00000990
IF(A(JPTR-1) .EQ. COMMA) GO TO 60	00010000
IF(NEXT(JPTR) .EQ. BLANK) RETURN	00010010
110 CALL ERROR(7,IDM1,IDM3,IDM2,IDM4)	00010020
RETURN	00010030
120 CALL ERROR(12,NXTID(1),NXTID(2),IDM3,IDM4)	00010040
RETURN	00010050
130 CALL ERROR(14,NXTID(1),NXTID(2),IDM3,IDM4)	00010060
RETURN	00010070
END	00010080



## Auxiliary Programs and Associated Data

### Program GRAPH

```

      DIMENSION INT(1000)
10  FORMAT(12I6)
20  FORMAT(2I6,A4,2X,I6,A4,2X,7I6/A4,2X,2I6,A4,2X,3I6,A4,2X,I6,A4,2X,
      $2I6/A4,2X,I6,A4,2X,3I6,A4,2X,2I6/A4,2X,2I6,A4,2X,8I6/A4,2X,I6,A4,
      $2X,I6,A4,2X,I6,A4,2X,3I6,A4,2X,A4,2X/3I6,A4,2X,3I6,A4,2X,3I6,A4,
      $2X/7I6,A4,2X,4I6/5I6,A4,2X,6I6/(12I6))
      READ 10,(INT(I),I=1,11)
      WRITE(4) (INT(I),I=1,11)
      READ 20,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT(1)
      WRITE(4) INT(1)
      READ 10,(INT(I),I=1,34)
      WRITE(4) (INT(I),I=1,34)
      STOP
      END

```

### Syntax Graph

1000	99	17	8	17	0	22	95	0	0	1	
(	359	60+	1-		2	103	103	67	3	3	4
*		72	5RL	0	669	483,		10CP		10000	349
=		7	480	795	7950P	20000	21	8*		9	42
*		18	480)	930	10	935	6	103	935	480	256
		12*	235=		13a		14	346	47*	OR	
	15	795	795?	16	391	411NT		17	322	359X	
	19	28	47	359	29	935	977+	110	67	318	318
	29	228	359	40	935/	130	3	72	11	72	60
	29	60	999	99	100	101	102	103	104	105	106
	107	62	109	111	112	113	114	115	116	117	118
	10	120	122	123	124	125	126	127	81	62	128
	131	133	134	135	136	137	157	138	93	140	95
	144	145	146	147	81	480	148	151	152	153	154
	156	110	157	159	93	114	95	160	164	165	166
	168	285	60	169	172	173	86	174	88	130	176
	133	114	72	95	180	185	186	187	188	189	516
	192	193	194	195	86	130	196	199	133	200	202
	204	95	205	207	208	209	210	211	212	103	126
											208



213	217	208	218	220	221	222	223	224	710	225	553
227	182	229	231	480	292	232	235	126	236	150	238
566	240	242	243	244	157	245	247	248	182	249	251
252	253	254	255	256	257	258	318	150	259	262	175
263	591	265	593	267	269	270	271	272	273	274	228
275	277	278	279	280	281	282	283	284	175	285	287
288	289	290	291	292	293	294	228	295	297	298	299
300	301	302	256	630	303	306	307	378	308	263	310
312	313	314	315	316	317	208	480	318	321	322	346
323	325	326	327	328	329	263	330	332	333	334	288
335	337	338	339	340	341	342	343	670	235	344	347
348	349	350	351	352	353	354	288	355	357	358	418
359	361	315	362	364	365	366	367	368	322	369	371
372	285	373	375	376	377	378	379	380	381	315	382
384	711	385	387	388	322	389	391	392	3	393	285
395	397	398	399	353	400	292	402	404	405	406	407
408	409	410	411	412	346	413	415	416	417	418	419
353	420	422	423	424	425	426	753	318	427	430	431
385	432	86	434	436	437	438	439	440	441	442	443
444	445	446	447	896	448	450	451	385	452	454	445
346	455	110	458	460	461	462	463	464	465	378	466
468	359	796	469	472	473	474	475	476	477	411	478
480	133	481	483	484	485	486	487	378	488	490	491
445	492	494	495	555	496	498	499	500	501	502	503
504	505	506	507	508	509	510	511	445	512	840	514
516	517	518	519	520	521	522	523	524	525	519	526
418	528	483	530	532	533	487	534	536	537	538	539
452	540	542	28	543	545	546	547	548	549	483	550
552	553	487	445	554	557	558	885	559	561	452	562
564	565	566	567	568	569	570	571	572	573	574	60
529	575	578	579	580	581	582	583	584	585	586	72
587	589	480	590	592	483	593	595	529	596	208	598
600	601	602	88	603	931	605	519	607	609	610	611
612	613	614	615	942	616	618	619	620	621	622	623
624	625	626	627	628	629	630	631	632	633	634	635
636	527	637	639	640	641	642	643	644	645	646	647
648	649	650	651	978	652	654	655	982	656	658	659
660	661	662	663	664	665	666	667	668	669	322	670
572	673	674	675	676	677	678	679	680	681	8	682
684	11	685	687	688	689	690	603	691	693	694	695
696	697	698	699	700	701	702	703	704	705	706	707
708	157	709	711	712	603	713	715	669	716	718	719
720	721	722	723	724	725	726	727	728	729	730	731
732	733	734	735	669	736	738	739	740	741	742	743
744	745	746	747	748	749	750	751	752	753	754	755
756	757	758	759	760	761	762	763	764	765	766	767
768	769	770	771	772	773	774	775	776	777	778	779
780	781	108	782	784	785	786	787	788	789	790	791
792	445	793	795	796	797	710	752	452	798	802	803
804	805	806	807	808	809	810	811	812	813	814	815
816	817	818	752	819	821	822	823	824	235	825	827
828	829	830	483	831	318	833	835	836	837	838	839
840	841	842	843	844	845	846	847	848	839	849	851
852	853	854	855	856	857	858	859	860	861	862	863
864	865	866	867	868	869	870	871	872	873	874	875
876	877	878	879	880	881	882	883	884	885	886	887
888	889	890	891	892	893	894	895	896	897	898	899
900	901	902	903	904	795	905	907	908	519	909	911
912	913	914	915	916	917	918	919	920	921	922	923
924	884	925	927	928	929	930	931	932	933	934	935
936	937	938	939	930	940	942	943	944	935	945	947
948	839	949	951	952	953	954	955	956	957	958	959
960	961	962	963	964	965	966	967	968	28	969	971
972	973	974	975	976	889	977	979	980	981	982	983
896	984	986	987	988	989	990	991	992	993	994	995
996	997	998	889								
43	89	4	683	6	66	8	149	710	657	21	24



12	89	617	17	268	19	800	44	28	23	783	13
26	228	130	29	820	31	241	89	266	35	305	37
38	36	130	15	4	345	129	191	89	4	130	378
50	428	94	89	54	471	56	515	445	67	42	61
560	710	839	65	606	5	346	69	653	12	896	73
686	15	4	78	93	535	10	33	889	15	84	555
86	577	89	87	88	10	8	89	597	386	93	98
519	108	100	101	102	103	104	105	106	107	109	96
111	93	112	113	114	115	116	117	118	119	120	122
4	123	124	125	126	127	128	131	1	175	133	4
134	135	136	137	138	140	182	142	133	144	452	145
146	147	148	151	7	130	152	153	154	155	156	157
159	208	160	164	157	228	89	165	166	167	168	169
172	315	5	173	174	176	89	179	89	29	180	185
208	555	11	89	186	187	188	189	190	192	20	193
194	195	196	199	10	603	200	202	235	203	204	205
207	16	208	209	210	211	212	213	217	5	89	110
218	220	452	221	222	223	224	225	227	483	229	25
231	133	232	235	130	235	236	238	3	240	157	242
30	243	244	245	247	89	248	249	251	150	252	253
254	255	256	257	258	259	262	89	3	263	265	519
267	80	269	16	270	271	272	273	274	275	277	256
278	279	280	281	282	283	284	285	287	3	288	289
290	291	292	293	294	295	297	285	298	299	300	301
302	303	306	346	34	307	308	310	411	312	29	313
314	315	316	317	318	321	1	130	322	323	325	445
326	327	328	329	330	332	292	333	334	335	337	263
338	339	340	341	342	343	344	347	59	2	348	349
350	351	352	353	354	355	357	889	358	359	361	89
362	364	256	365	366	367	368	369	371	208	372	373
375	89	376	377	378	379	380	381	382	384	285	385
387	51	388	389	391	126	392	393	395	21	397	13
398	399	400	402	208	404	1	405	406	407	408	409
410	411	412	413	415	89	416	417	418	419	420	422
89	423	424	425	426	427	430	49	5	431	432	434
29	436	529	437	438	439	440	441	442	443	444	445
446	447	448	450	935	451	452	454	418	455	458	95
13	460	288	461	462	463	464	465	466	468	89	469
472	4	53	473	474	475	476	477	478	480	378	481
483	263	484	485	486	487	488	490	13	491	492	494
29	495	496	498	603	499	500	501	502	503	504	505
506	507	508	509	510	511	512	514	480	516	55	517
518	519	520	521	522	523	524	525	526	528	62	530
3	532	669	533	534	536	76	537	538	539	540	542
89	543	545	47	546	547	548	549	550	552	710	553
554	557	83	6	558	559	561	60	562	564	5	565
566	567	568	569	570	571	572	573	574	575	578	81
85	579	580	581	582	583	584	585	586	587	589	529
590	592	7	593	595	14	596	598	95	600	110	601
602	603	605	669	607	64	609	89	610	611	612	613
614	615	616	618	40	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635	636	637
639	5	640	641	642	643	644	645	646	647	648	649
650	651	652	654	68	655	656	658	90	659	660	661
662	663	664	665	666	667	668	669	670	672	385	673
674	675	676	677	678	679	680	681	682	684	3	685
687	72	688	689	690	691	693	89	694	695	696	697
698	699	700	701	702	703	704	705	706	707	708	709
711	182	712	713	715	9	716	718	208	719	720	721
722	723	724	725	726	727	728	729	730	731	732	733
734	735	736	738	88	739	740	741	742	743	744	745
746	747	748	749	750	751	752	753	754	755	756	757
758	759	760	761	762	763	764	765	766	767	768	769
770	771	772	773	774	775	776	777	778	779	780	781
782	784	22	785	786	787	788	789	790	791	792	793
795	114	796	797	798	802	89	18	487	803	804	805
806	807	808	809	810	811	812	813	814	815	816	817



818	819	821	28	822	823	824	825	827	322	828	829
830	831	833	752	835	40	836	837	838	839	840	841
842	843	844	845	846	847	848	849	851	884	852	853
854	855	856	857	858	859	860	861	862	863	864	865
866	867	868	869	870	871	872	873	874	875	876	877
878	879	880	881	882	883	884	885	886	887	888	889
890	891	892	893	894	895	896	897	898	899	900	901
902	903	904	905	907	15	908	909	911	62	912	913
914	915	916	917	918	919	920	921	922	923	924	925
927	930	928	929	930	931	932	933	934	935	936	937
938	939	940	942	977	943	944	945	947	3	948	949
951	4	952	953	954	955	956	957	958	959	960	961
962	963	964	965	966	967	968	969	971	710	972	973
974	975	976	977	979	89	980	981	982	983	984	986
89	987	988	989	990	991	992	993	994	995	996	997
998	999	99	3								
43	2	4	3	6	5	8	7	9	90	11	13
12	14	40	17	16	19	18	44	21	23	22	24
26	25	110	29	28	31	30	32	80	35	34	717
38	37	191	15	41	59	1	20	45	46	276	52
50	49	94	324	54	53	56	55	394	58	42	61
60	737	311	65	64	66	459	69	68	433	370	73
72	139	158	78	77	76	493	33	82	178	84	83
86	85	89	88	87	10	241	92	95	51	93	98
97	96	0	0	0	0	0	0	0	0	0	108
0	27	0	0	0	0	0	0	0	0	0	0
121	0	0	0	0	0	0	0	129	130	0	132
0	0	0	0	0	0	74	0	141	0	143	0
0	0	0	0	149	150	0	0	0	0	0	0
0	75	0	0	161	162	163	0	0	0	0	0
0	170	171	0	0	0	175	0	177	81	0	0
181	182	183	184	0	0	0	0	0	0	39	0
0	0	0	0	197	198	0	0	201	0	0	0
0	206	0	0	0	0	0	0	0	214	215	216
0	0	219	0	0	0	0	0	0	226	0	228
0	230	0	0	233	234	0	0	237	0	239	0
91	0	0	0	0	246	0	0	0	250	0	0
0	0	0	0	0	0	0	260	261	0	0	264
0	266	0	268	0	0	0	0	0	0	0	47
0	0	0	0	0	0	0	0	0	286	0	0
0	0	0	0	0	0	0	296	0	0	0	0
0	0	0	304	305	0	0	0	309	0	63	0
0	0	0	0	0	0	319	320	0	0	0	48
0	0	0	0	0	0	331	0	0	0	0	336
0	0	0	0	0	0	0	0	345	346	0	0
0	0	0	0	0	0	0	356	0	0	0	360
0	0	363	0	0	0	0	0	0	71	0	0
0	374	0	0	0	0	0	0	0	0	383	0
0	386	0	0	0	390	0	0	0	57	0	396
0	0	0	0	401	0	403	0	0	0	0	0
0	0	0	0	0	414	0	0	0	0	0	0
421	0	0	0	0	0	0	428	429	0	0	0
70	0	435	0	0	0	0	0	0	0	0	0
0	0	0	0	449	0	0	0	453	0	0	456
457	0	67	0	0	0	0	0	0	0	467	0
0	470	471	0	0	0	0	0	0	0	479	0
0	482	0	0	0	0	0	0	489	0	0	0
79	0	0	0	497	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	513	0	515	0
0	0	0	0	0	0	0	0	0	0	527	0
529	0	531	0	0	0	535	0	0	0	0	0
541	0	0	544	0	0	0	0	0	0	551	0
0	0	555	556	0	0	0	560	0	0	563	0
0	0	0	0	0	0	0	0	0	0	0	576
577	0	0	0	0	0	0	0	0	0	0	588
0	0	591	0	0	594	0	0	597	0	599	0
0	0	0	604	0	606	0	608	0	0	0	0



0	0	0	0	617	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	638	0	0	0	0	0	0	0	0	0	0
0	0	0	0	653	0	0	0	657	0	0	0
0	0	0	0	0	0	0	0	0	0	671	0
0	0	0	0	0	0	0	0	0	0	683	0
0	686	0	0	0	0	0	692	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	710	0	0	0	714	0	0	36	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	62	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	783	0	0	0	0	0	0	0	0	0
0	794	0	0	0	0	799	800	801	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	820	0	0	0	0	0	826	0	0
0	0	0	832	0	834	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	850	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	906	0	0	0	910	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	926	0	0	0	0	0	0	0	0	0	0
0	0	0	0	941	0	0	0	0	946	0	0
0	950	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	970	0	0
0	0	0	0	0	978	0	0	0	0	0	0
985	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1000	0	0	0	0	0	0	0	0
96	76	99	97	99	97	96	96	76	97	76	97
99	76	97	99	97	96	96	99	76	99	97	236
99	77	72	96	96	99	97	76	97	99	97	232
99	97	72	99	77	97	96	97	76	77	72	72
99	97	99	72	99	97	99	97	72	76	99	99
97	72	72	99	97	236	72	99	97	73	72	99
97	73	73	96	76	96	72	99	72	73	96	96
96	96	96	236	96	99	73	76	96	97	96	96
76	96	0	0	0	0	0	0	0	0	0	236
0	76	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	236	76	0	77
0	0	0	0	0	0	76	0	76	0	76	0
0	0	0	0	236	76	0	0	0	0	0	0
0	76	0	0	76	76	76	0	0	0	0	0
0	76	77	0	0	0	76	0	76	76	0	0
76	76	77	76	0	0	0	0	0	0	236	0
0	0	0	0	77	76	0	0	76	0	0	0
0	77	0	0	0	0	0	0	0	77	76	76
0	0	76	0	0	0	0	0	0	76	0	236
0	76	0	0	76	76	0	0	77	0	76	0
236	0	0	0	0	76	0	0	0	76	0	0
0	0	0	0	0	0	0	76	77	0	0	76
0	236	0	236	0	0	0	0	0	0	0	76
0	0	0	0	0	0	0	0	0	77	0	0
0	0	0	0	0	0	0	76	0	0	0	0
0	0	0	76	236	0	0	0	76	0	76	0
0	0	0	0	0	0	77	76	0	0	0	76
0	0	0	0	0	0	76	0	0	0	0	76
0	0	0	0	0	0	0	0	236	77	0	0
0	0	0	0	0	0	0	76	0	0	0	76
0	0	76	0	0	0	0	0	0	76	0	0
0	76	0	0	0	0	0	0	0	0	76	0
0	236	0	0	0	76	0	0	0	76	0	77



0	0	0	0	76	0	77	0	0	0	0	0
0	0	0	0	0	76	0	0	0	0	0	0
76	0	0	0	0	0	0	236	77	0	0	0
76	0	76	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	76	0	0	76
77	0	76	0	0	0	0	0	0	0	76	0
0	77	236	0	0	0	0	0	0	0	76	0
0	76	0	0	0	0	0	0	77	0	0	0
76	0	0	0	76	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	76	0	236	0
0	0	0	0	0	0	0	0	0	0	76	0
77	0	76	0	0	0	236	0	0	0	0	0
76	0	0	76	0	0	0	0	0	0	76	0
0	0	236	77	0	0	0	236	0	0	77	0
0	0	0	0	0	0	0	0	0	0	0	76
236	0	0	0	0	0	0	0	0	0	0	76
0	0	77	0	0	77	0	236	0	76	0	0
0	0	0	76	0	236	0	76	0	0	0	0
0	0	0	0	236	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	77	0	0	0	0	0	0	0	0	0	0
0	0	0	0	236	0	0	0	236	0	0	0
0	0	0	0	0	0	0	0	0	0	76	0
0	0	0	0	0	0	0	0	0	0	236	0
0	236	0	0	0	0	0	76	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	76	0	0	0	77	0	0	76	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	236	0	0	0	0	0	0	0	0	0
0	76	0	0	0	0	76	236	76	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	236	0	0	0	0	0	76	0	0
0	0	0	76	0	76	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	77	0	0	76	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	76	0	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	0	76	0	0
0	77	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	76	0	0
0	0	0	0	0	76	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0	0	0
0	0	0	77	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
8	391	982	349	942	516	553	591	630	670	711	753
796	840	885	931	978	42	11	48	68	89	111	675
154	642	137	593	108	566	19	29	228	483		



# Program SESLIST

```

INTEGER TYPE,CLASS,CLASS2,BLKSI(100),BLKTY(100)
DIMENSION ISUBLT(4,200),INTFAC(600),IARGS(200),NAME(2)
EQUIVALENCE (BLKTY(1),IARGS(1)),(BLKSI(1),IARGS(101))
EQUIVALENCE (ITEMP,RTEMP)
NLIST=0
IPTR=1
1 READ(5,5,END=40) NAME,NARGS,TYPE,CLASS,ISIZE
5 FORMAT(A4,A2,1X,3I2,1X,I6)
IF(ISIZE .LT. 1) ISIZE=0
NLIST=NLIST+1
IVAR=0
ISUBLT(1,NLIST)=NAME(1)
ISUBLT(2,NLIST)=NAME(2)
IF(NARGS .NE. -1) GO TO 7
NARGS=1
IVAR=1
7 NARG2=NARGS
CLASS2=CLASS
CALL SHIFTL(CLASS2,22)
CALL SHIFTL(NARG2,26)
CALL SHIFTL(TYPE,19)
RTEMP=OR(ISIZE,CLASS2)
RTEMP=OR(RTEMP,NARG2)
RTEMP=OR(RTEMP,TYPE)
ISUBLT(3,NLIST)=ITEMP
ISUBLT(4,NLIST)=0
IF(NARGS .EQ. 0) GO TO 1
CALL SHIFTL(IVAR,18)
RTEMP=OR(ISUBLT(3,NLIST),IVAR)
ISUBLT(3,NLIST)=ITEMP
ISUBLT(4,NLIST)=IPTR
IF(CLASS .EQ. 7) GO TO 25
JPTR=IPTR
INC=1+(NARGS-1)/3
IPTR=IPTR+INC
NDPTR=IPTR-1
NPARAM=3*NARGS
READ(5,10) (IARGS(I),I=1,NPARAM)
10 FORMAT(20(3I1,1X))
KOUNT=0
DO 22 I=JPTR,NDPTR
ITEMP=0
NSHIFT=32
DO 20 K=1,9
KOUNT=KOUNT+1
IF(KOUNT .GT. NPARAM) GO TO 22
NSHIFT=NSHIFT-3
ISHIFT=NSHIFT
IF(MOD(KOUNT,3) .EQ. 0) ISHIFT=NSHIFT+1
CALL SHIFTL(IARGS(KOUNT),ISHIFT)
20 RTEMP=OR(RTEMP,IARGS(KOUNT))
22 INTFAC(I)=ITEMP
GO TO 1
25 JPTR=IPTR
IPTR=IPTR+NARGS
NDPTR=IPTR-1
READ(5,27) (BLKSI(I),BLKTY(I),I=1,NARGS)
27 FORMAT(10(I6,1X,I1))
KOUNT=0
DO 30 I=JPTR,NDPTR
KOUNT=KOUNT+1
CALL SHIFTL(BLKSI(KOUNT),15)
CALL SHIFTL(BLKTY(KOUNT),12)
RTEMP=OR(BLKSI(KOUNT),BLKTY(KOUNT))
30 INTFAC(I)=ITEMP
GO TO 1
40 WRITE(4) NLIST,NDPTR
WRITE(4) ((ISUBLT(I,J),I=1,4),J=1,NLIST)
WRITE(4) (INTFAC(I),I=1,NDPTR)
WRITE(6,50) NLIST
50 FORMAT(//42X,37H NEW LIST HAS BEEN CREATED CONTAINING,I4,6H NAMES)
STOP
END

```



# Basic Interface Definition File

```

ABS      1 1 4
102
AIMAG    1 1 4
202
AINT     1 1 4
102
ALOG     1 1 4
102
ALOG10   1 1 4
102
AMAX0    -1 1 4
402
AMAX1    -1 1 4
102
AMIN0    -1 1 4
402
AMIN1    -1 1 4
102
AMOD      2 1 4
102 102
ATAN      1 1 4
102
ATAN2     2 1 4
102 102
CABS      1 1 4
202
CCOS      1 2 4
202
CEXP      1 2 4
202
CLOG      1 2 4
202
CMPLX     2 2 4
102 102
CONJG     1 2 4
202
COS       1 1 4
102
CSIN      1 2 4
202
CSQRT     1 2 4
202
DABS      1 3 4
302
DATAN     1 3 4
302
DATAN2    2 3 4
302 302
DBLE      1 3 4
102
DCOS      1 3 4
302
DEXP      1 3 4
302
DIM        2 1 4
102 102

```

```

DLOG      1 3 4
302
DLOG10    1 3 4
302
DMAX1     -1 3 4
302
DMIN1     -1 3 4
302
DMOD       2 3 4
302 302
DSIGN      2 3 4
302 302
DSIN       1 3 4
302
DSQRT      1 3 4
302
EXP        1 1 4
102
FLOAT      1 1 4
402
IABS       1 4 4
402
IDIM       2 4 4
402 402
IDINT      1 4 4
302
IFIX       1 4 4
102
INT         1 4 4
102
ISIGN      2 4 4
402 402
MAX0       -1 4 4
402
MAX1       -1 4 4
102
MIN0       -1 4 4
402
MIN1       -1 4 4
102
MOD         2 4 4
402 402
REAL       1 1 4
202
SIGN       2 1 4
102 102
SIN        1 1 4
102
SNGL       1 1 4
302
TAN         1 1 4
102
TANH       1 1 4
102

```



INITIAL DISTRIBUTION

Copies

10	NAVSEA PMS304-32 White
2	NAVSEA PMS405-40 Cuthbert
12	DDC

CENTER DISTRIBUTION

1	18/1809
1	1802.2 Frenkiel
1	1802.4 Theilheimer
1	1809.3 D. Harris (Central Depository, CMLD)
1	182 Camara
1	1826 Culpepper
30	1826 Wybraniec
1	184 Lugt
1	185 Corin
1	186 Sulit
1	189 Gray
1	1890 Taylor
30	5214.1 Reports Distribution
1	522

Microfiche copies

30	1826 Wybraniec
----	----------------



DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.